Federal University of Espírito Santo

Graduate Program in Computer Science

Ghent University

Faculty of Economics and Business Administration

MARIA DAS GRAÇAS DA SILVA TEIXEIRA

An Ontology-Based Process for Domain-Specific Visual Language Design

Doctoral Dissertation

Vitória-ES, Brazil

August, 2017

MARIA DAS GRAÇAS DA SILVA TEIXEIRA

An Ontology-Based Process for Domain-Specific Visual Language Design

Doctoral dissertation submitted to Graduate Programs from the Federal University of Espírito Santo and the Ghent University as partial requirement for obtaining the degree of Doctor of Computer Science (UFES) and Doctor of Applied Economic Sciences (UGent), under the guidance of Dr. Giancarlo Guizzardi and co-supervision of Dr. Frederik Gailly and Dr. Ricardo de Almeida Falbo.

Vitória-ES, Brazil

August, 2017

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

-
266o

Teixeira, Maria das Graças da Silva, 1977-
An ontology-based process for domain-specific visual language design / Maria das Graças da Silva Teixeira. – 2017.
247 f. : il.

Orientador: Giancarlo Guizzardi.
Coorientadores: Ricardo de Almeida Falbo, Frederik Gailly.
Tese (Doutorado em Ciência da Computação, Doutorado em Ciências Econômicas Aplicadas) – Universidade Federal do Espírito Santo, Centro Tecnológico. Ghent University, Faculty of Economics and Business Administration.

1. Linguagem de programação (Computadores) – Sintaxe. 2. Engenharia de Linguagem. 3.Linguagem de modelagem visual específica de domínio. 4. OntoUML (Linguagem de modelagem padrão). 5. Física das notações (PoN). 6. Ontologia fundamental unificada (UFO). 7. Notação visual. I. Guizzardi, Giancarlo. II. Falbo, Ricardo de Almeida. III. Gailly, Frederik. IV. Universidade Federal do Espírito Santo. Centro Tecnológico. V. Ghent University, Faculty of Economics and Business Administration. VI. Título.

CDU: 004

MARIA DAS GRAÇAS DA SILVA TEIXEIRA

An Ontology-Based Process for Domain-Specific Visual Language Design

Doctoral dissertation submitted to the Graduate Programs from the Federal University of Espírito Santo and the Ghent University as partial requirement for obtaining the degree of Doctor of Computer Science (UFES) and Doctor of Applied Economic Sciences (UGent).

Approved on August 17, 2017.

Giancarlo Guizzardi, Ph.D.
Federal University of Espírito Santo, Brazil
(supervisor)

Ricardo de Almeida Falbo, DSc.
Federal University of Espírito Santo, Brazil
(co-supervisor)

Frederik Gailly, Ph.D.
Ghent University, Belgium
(co-supervisor)

João Paulo Andrade Almeida, Ph.D.
Federal University of Espírito Santo, Brazil
(internal member, UFES)

Maria Luiza Machado Campos, Ph.D.
Federal University of Rio de Janeiro, Brazil
(external member)

Geert Poels, Ph.D.
Ghent University, Belgium
(internal member, UGent)

Amy Van Looy, Ph.D.
Ghent University, Belgium
(internal member, UGent)

“I know things and I drink wine.”

Tyrion Lanister, Game of Thrones [George R.R. Martin]

Dedication

To my parents, João Cornélio and Maria Emília, for being my guides since the past way
To my sisters, Karen Cristina and Ana Karyna, for being my companions in the present way
To my nephews, João Felipe and Marina, for being my heart in the future way

Acknowledgments

This is the final moment of a doctoral research. The moment to thank. I am grateful to have been able to complete this phase in my professional and personal journey. Most important, I am grateful to all those who have somehow been part of this journey.

To the professionals I met during the doctoral program and at the events I had the opportunity to attend. A professional network is under construction. I am appreciating it since the first moment.

To the Federal University of Espirito Santo (UFES) and the Ghent University (UGent) that accepted me as a doctoral student and provided me with all the necessary infrastructure and professionals. Also, to the Laboratory of Applied Ontology (LOA) in which I researched with during a technical stage.

To CNPq, CAPES and FAPES, which gave me the necessary financial support so that I could establish a partnership with UGent and LOA.

To the professors of the PPGI/UFES who from the beginning of the doctoral program presented me to a new world of research possibilities and were always open to discussion.

To the members of DCEL/UFES, the department in which I work. The fact that we are a united and promising group of professors allows us to evolve ourselves more and more, and over time we will make a difference in our fields of work.

To the colleagues of the research groups of NEMO/UFES and MIS/UGent. Finding out what it is to research with you all has made this adventure much more interesting and productive.

To the members of the examination board, who showed me several possibilities to refine the research and to improve the text.

To my supervisors, Giancarlo Guizzardi, Ricardo Falbo and Frederik Gailly. Thank you for guiding me, for sharing your knowledge, for forcing me to move on, for making me see new possibilities and also failures, for so many discussions and joint work. I can only say "when I grow up, I want to be like you". I wish we could maintain our partnership for a long time. I still have a lot to learn from you.

To the friends, who understood and supported me during this long journey. In particular, to Cristine Griffo, who started as a doctoral student of the same program and today is a great friend and research colleague. May many meetings, travels, laughter and work come.

To my family, João Cornélio, Maria Emília, Karen, Karyna, Marina, João Felipe, Roberto, Carlos. The best family I could dream of. The light and support of my life.

To God, because all we are and we have, we owe to Him.

A doctorate is not a solitary, easy and short-term work, but a long and hard journey that can only be traveled thanks to all the support, company and lessons that we have received in such a journey. I acknowledge this and register here the thanks to you who accompanied me on such a journey ... Obrigada! Thank you! Dank u wel! Grazie mille!

SUMMARY IN ENGLISH

In Conceptual Modeling, there has been a growing interest in Domain-Specifc Visual Modeling Languages (DSVML) and the support they provide for problem domain understanding and communication between stakeholders. Thus, it is important to provide guidelines for designing DSVMLs. For many years, the research focus has been on abstract syntax, whilst concrete syntax has received lower attention. This is unfortunate, because the visual syntax has a significant impact on the efficiency of the communication and problem solving capabilities of conceptual models.

One of the most widespread work for analysis and design of visual aspects of modeling languages is the Physics of Notations (PoN). PoN defines a set of principles used for analyzing and designing cognitively effective visual notations. However, PoN has shortcomings, as: (i) it lacks a method to apply its principles; (ii) the design of symbols does not systematically create symbols that reflect entities in the real-world.

In this work, we present the *Physics of Notations Systematized (PoN-S)* for solving shortcoming (i). PoN-S establishes an ordered set of design activities and suggests when to apply the PoN principles. Also, it proposes groups of PoN principles.

Another way for improving the DSVML quality involves the application of ontological theories, but can ontologies be successfully applied in the design of concrete syntax as it has been applied for abstract syntax? Guizzardi (2013) proposes some Unified Foundational Ontology (UFO)-based guidelines that helps in the design of visual notations. However, also these guidelines have some shortcomings: (iii) The guidelines are isolated guidelines and not part of a design process; (iv) The set of considered constructs and, thus, the set of ontological distinctions considered is restricted; (v) The ontological guidelines are restricted to be applied in the establishment of the DSVML symbols activity.

To solve shortcomings (ii) till (v) we combined the UFO-based guidelines with PoN-S, giving rise to the *Physics of Notations Ontologized and Systematized (PoNTO-S)*. PoNTO-S is a systematic design process of DSVML concrete syntax that connects the concrete syntax and the real-world (i.e., ontological) semantics.

This research is a Design Science project with different iterations, each one producing their own artifacts. The design problem faced is the design of DSVML concrete syntaxes. The artifacts, PoN-S and PoNTO-S, were improvements upon two existing solutions of the problem: PoN theory and UFO-based guidelines. These artifacts can be classified as design theories, as they are design processes. Also, we invested in empirical studies. We performed exploratory studies to support the indications collected in the literature review and guide some decisions. After developing releases of PoN-S and PoNTO-S we applied one experiment and two case studies that generated evidences for us to concluded that PoN-S and PoNTO-S are helpful in the language development, and that both approaches can evolve, given rise to more useful approaches.

Keywords: Concrete Syntax, Design Process, Domain-Specific Visual Modeling Language, Language Engineering, OntoUML, PoN, UFO, Visual Notation

RESUMO EM PORTUGUÊS

Em Modelagem Conceitual, tem ocorrido um interesse crescente em Linguagens de Modelagem Visuais Específicas de Domínio (*Domain-Specific Visual Modeling Languages* (DSVMLs)) e no suporte que elas provêem para compreensão do domínio de um problema e comunicação entre modeladores e interessados. Assim, é importante providenciar diretrizes para o design de DSVMLs. Por muitos anos, o foco de pesquisa tem sido na sintaxe abstrata, enquanto a sintaxe concreta tem recebido menor atenção. Isso é um infortúnio, pois a sintaxe visual impacta significativamente a capacidade de comunicação e de resolução de problemas de modelos conceituais

Um dos trabalhos mais disseminados para análise e design de aspectos visuais de linguagens de modelagem é a Física das Notações (PoN). PoN define um conjunto de princípios usado para analisar e projetar notações visuais cognitivamente eficientes. Contudo, PoN tem lacunas, tais como: (i) Falta um método para aplicar seus princípios; (ii) O design de símbolos não cria sistematicamente símbolos que refletem entidades do mundo real.

Nesta pesquisa, nós apresentamos a *Física das Notações Sistematizada (PoN-S)* para resolver a lacuna (i). PoN-S estabelece um conjunto ordenado de atividades de design e sugere quando aplicar os princípios de PoN. Ela também propõe grupos de princípios de PoN.

Outra maneira de melhorar a qualidade de DSVMLs é a aplicação de teorias ontológicas, mas ontologias podem ser aplicadas com sucesso no design de sintaxes concretas como tem sido aplicadas para a sintaxe abstrata? Guizzardi (2013) propoem diretrizes ontológicas baseadas na Ontologia de Fundamentação Unificada (UFO) para auxiliar no design de notações visuais. Contudo, tais diretrizes também tem lacunas: (iii) São diretrizes isoladas e não parte de um processo de design; (iv) O conjunto de distinções ontológicas, é restrito; (v) As diretrizes ontológicas são restritas a serem aplicadas no estabelecimento dos símbolos de uma DSVMLs.

Para resolver as lacunas (ii) até (v) nós combinamos as diretrizes ontológicas baseadas em UFO com PoN-S, originando a *Física das Notações Ontologizada e Sistematizada (PoNTO-S)*. PoNTO-S é um processo de design sistematizado para sintaxes concretas de DSVMLs que conecta a sintaxe concreta com o significado do mundo-real (isto é, o significado ontológico).

Este projeto é um processo de Design Science com diferentes iterações, cada uma produzindo artefatos próprios. O problema de design é o design de sintaxes concretas de DSVMLs. Os artefatos são melhorias de duas soluções existentes: PoN e diretrizes ontológicas baseadas em UFO. PoN-S e PoNTO-S são classificados como teorias de design, visto serem processo de design. Nós também investimos em estudos empíricos. Foram executados estudos exploratórios para dar suporte as indicações coletadas durante a revisão de literatura e guiar algumas decisões. Após desenvolver versões de PoN-S e PoNTO-S nós aplicamos novos estudos empíricos que geraram evidências para concluirmos que PoN-S e PoNTO-S são utéis, e que tais abordagens podem evoluir, dando origem a abordagens ainda mais úteis.

NEDERLANDSTALIGE SAMENVATTING

In het domein van de conceptuele modellering wordt er steeds meer aandacht besteed aan visuele domeinspecifieke modelleertalen en hoe deze talen ondersteuning kunnen bieden bij het representeren van een bepaald domein voor verschillenden belanghebbenden. Bijgevolg is er een absolute noodzaak aan richtlijnen die men kan volgen bij het ontwikkelen van deze domeinspecifieke modelleertalen. Bestaand onderzoek voorziet een aantal richtlijnen maar deze focussen meestal op de abstracte syntax van deze talen en niet op de visuele aspecten (concrete syntax) van deze talen. Er is nochtans een absolute noodzaak aan richtlijnen specifiek voor de ontwikkeleng van de concrete syntax want deze heeft een significante impact op de efficiëntie van de communicatie en probleemoplossende eigenschappen van de met deze talen ontwikkelde conceptuele modellen.

De meest gebruikte theorie voor de evaluatie van de concrete syntax van een visuele modelleertaal is de Physics of Notations(PoN). PoN definieert een verzameling principes die men kan gebruiken voor de analyse en ontwerp van een cognitief effectieve visuele notatie voor een modelleertaal. PoN heeft echt ook een aantal tekortkomingen: i) het bevat geen methode die aangeeft hoe de principes moeten gebruikt worden en ii) het helpt niet bij het ontwikkelen van symbolen die overeenstemmen met het domein.

In dit PhD project wordt de Physics of Notations Systematized (PoN-S) ontwikkeld en voorgesteld als een oplossing voor de eerste tekortkoming van PoN. PoN-S voorziet een sequentiële set van activiteiten en geeft voor elke activiteit aan welk principe moet worden gebruikt. Bovendien voorziet het ook een groepering voor de verschillende principes die de gebruiker moet helpen bij het gebruik.

De tweede tekortkoming wordt in dit PhD project opgelost door gebruik te maken van foundational ontologies. Foundational ontologies worden gebruikt voor het verbeteren van de kwaliteit van zowel de abstracte syntax van een modelleertaal als ook voor het rechtstreeks verbeteren van het conceptueel model. In dit doctoraat wordt het onderzoek van Guizzardi (2013) en meer specifiek het onderzoek rond UFO gebaseerde ontologische richtlijnen gecombineerd met de eerder ontwikkelde verbetering van PoN. Dit resulteert in de Physics of Notations Ontologized and Systematized (PoNTO-S), een systematisch ontwikkelingsproces voor de concrete syntax van visuele modelleertalen waarbij ook rekening wordt gehouden met de ontologische betekenis van de abstracte syntax.

Het onderzoek dat uitgevoerd werd in het kader van dit PhD project stemt overeen met een Design Science project met verschillende iteraties die resulteren in verschillende Design Science artefacten die ook werden geëvalueerd. Na de ontwikkeling van PoN-S en PoNTO-S werd er één labo experiment uitgevoerd en werden de artefacten ook deels geëvalueerd door gebruik te maken van twee case studies. Deze studies tonen aan dat PoN-D en PonTO-S nuttig zijn tijdens de ontwikkeling van visuele domeinspecifeke modelleertalen.

Table List

Figure List

Summary

Chapter 1. Introduction

"While SE has developed mature methods for evaluating and designing semantics, it lacks equivalent methods for visual syntax." (MOODY, 2009)

In this chapter, we present the context in which this work is inserted, the motivation to develop the research, the research problem identified, a formulated hypothesis to solve the problem and the established goals to implement the hypothesis solution, the methodological aspects planned to guide the research to achieve the goals, a list of publications that presented the research aspects, and, finally, the strutucture of the dissertation text.

1.1 Context and Motivation

Communication is an intrinsic human characteristic and a fundamental factor in all aspects of society. According to Schramm (1955), communication has three elements: source, destination and message. The message is the information we want to share. The source identifies the one that encodes the message to be sent to the destination, which, in turn, is the one who receives the message and decodes it, thus closing the communication cycle. The message may have a variety of formats (signs) and these formats should be capable of meaningful interpretation allowing the message to be appropriately shared between source and destination. The communication process should establish a common understanding between the source and the destination. This implies that both encoding and decoding process of the message should use a common language, which can be interpreted by both source and destination.

In Software Engineering, conceptual models are used for communication between stakeholders. A ***conceptual model*** is a representation of aspects of the physical and social world to support communication, problem solving, and meaning negotiation between humans (MYLOPOULOS, 1992). A conceptual model, as a communication mean, must, therefore, be in accordance with the precepts of communication process, as the ones previously presented. The content of the model is the message. The sign is the model representation (its format), and such a format is usually graphical (diagrammatic). Source and destination are the stakeholders involved in the communication. As a conceptual model is elaborated using a ***Conceptual Modeling Language (CML)***, the "common understanding" can be mediated by this language, which acts either during coding and decoding processes.

To reinforce the association between communication and language, Brandone *et al.* (2006) indicate that content refers to the language semantics, that is, concepts and ideas that can be codified, while format is the way in which meaning is represented.

Regarding modeling languages, while the language ***abstract syntax*** identifies the modeling elements, characterizing the meaning involved in the semantics, the ***concrete syntax*** establishes how such elements should be represented by graphics or textual elements (BAAR, 2008), that is, it identifies how the semantics could be visualized. Thus, a direct association can be made: the model content is characterized in terms of the language abstract syntax, while the format is identified by the concrete syntax. Figure 1 highlights the interrelation between the basic components of Communication and Conceptual Modeling.

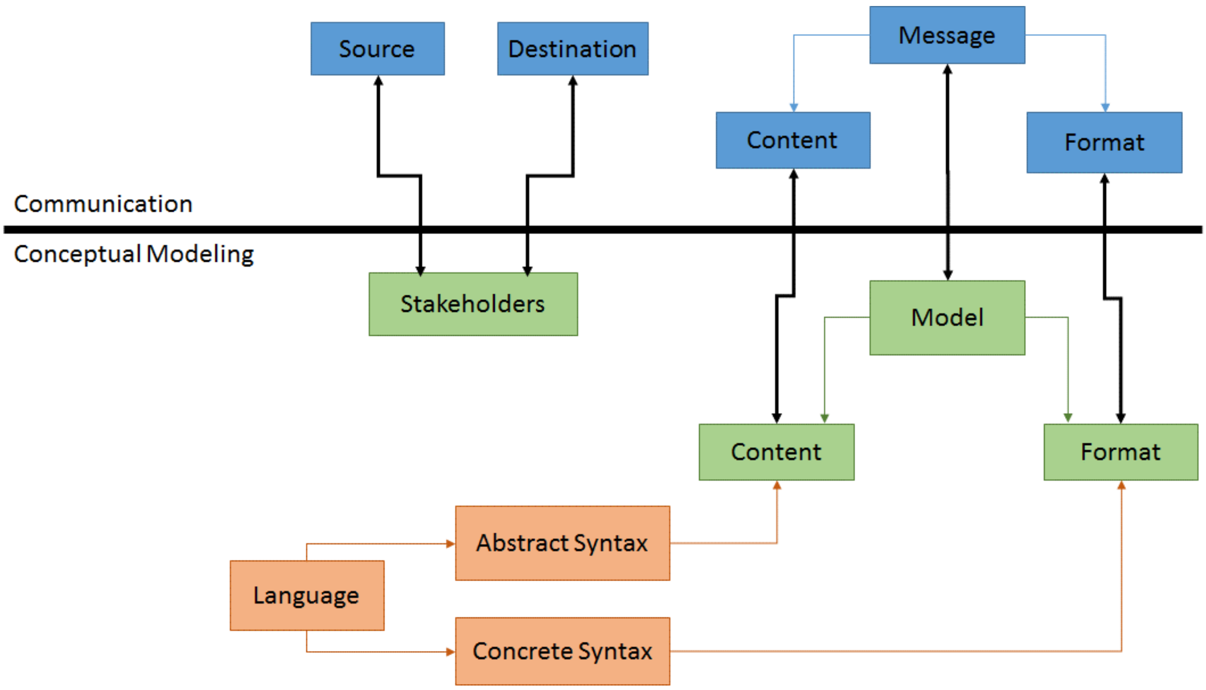


Figure 1: Relation between the basic components of Communication and Conceptual Modeling

Since the goal is that the message encoded by the source is decoded by the destination in the proper way, the language used for encoding and decoding the message should be of quality. Thus, if we want that a conceptual model fulfills its communicating role, then the Conceptual Modeling language in which the model is coded and decoded should be of quality and therefore the language abstract and concrete syntaxes should be of quality.

We emphasize that we are not assuming that the quality of a modeling language is the only factor that impacts on the quality of the communication process. Our claim is only that, in Conceptual Modeling, the quality of the language impacts on the communication capability of the conceptual model.

This reflection leads us to the central question of this research: How to increase the quality of domain-specific modeling languages, and consequently of their communication capacity?

Visual notations play a critical role in Software and Systems Engineering. However, when designing notations, most of the effort is commonly spent on defining the abstract syntax and the semantic, whereas the design of effective graphical syntaxes tends to be made in an *ad hoc* and subjective way, without proper guidance from theoretical frameworks based on empirical evidence. This lack of attention and rigour negatively impacts the cognitive effectiveness of notations, i.e., the speed, accuracy, and ease with which information can be understood (GENON *et al.*, 2010).

The effectiveness of software models depends on a number of communication qualities, such as: production cost, comprehension degree, speed of 'fall' (loss of synchrony with the represented content) and slope of the learning curve (ARANDA *et al.*, 2007). If a language is deficient in several of these qualities, then it does not matter whether it has a high power of expressiveness and a well-formalized semantics. It will not be properly used for communication purposes (ARANDA *et al.*, 2007). Thus, even if a language has a good quality abstract syntax, if it has serious shortcomings in concrete syntax, it may simply not be used by the community. Several works point out the need to invest in both abstract and concrete syntaxes (MORRIS; SPANOUDAKIS, 2001) (GENON *et al.*, 2010) (BOBKOWSKA, 2013).

In this work, we are interested in improving the concrete syntax of DSMLs as a means to improve the quality of conceptual modeling languages and as consequence to turn easier the use of modeling languages by the user community.

1.2 Research Problem and Hypothesis

Considering the content and motivation described, we formulated the following research problem: ***How to improve the quality of the concrete syntax of domain-specific visual modeling languages (DSVMLs)***?

After executing a literature review and performing some exploratory empirical studies we developed the following hypothesis as an answer to the research problem: ***To improve the quality of the concrete syntax of DSVMLs, we should use a systematic process that consistently applies Information Visualization and Ontological theories in the design of DSVML concrete syntaxes***.

Information Visualization theories are an obvious basis for the design of DSVML concrete syntaxes, due to its visual aspect. Ontological theories can guide both the design of DSVML abstract syntax and the design of the corresponding concrete syntax, as both syntaxes should do semantic considerations and the semantic schema associated to the DSVML can be described through an ontology.

When designing DSVML concrete syntaxes, we need to take Information Visualization theories into account, because the concrete syntax elements are the language components that we literally “see”. However, it is recognized that little attention has been given to such aspect. For instance, Figl (2012) claims that there is a lack of scientific justifications for the choices related to the symbols of notations. Also, Moody (2009) points out that rarely the issues involving the modeling language visual syntax are discussed.

Ontological theories have been used mainly as a basis for evaluating and (re)designing modeling languages. This is the case, for instance, of the Unified Foundational Ontology - UFO (GUIZZARDI, 2005), in works such as (GUIZZARDI, 2013) and (AZEVEDO *et al.*, 2013), as well as the BWW ontology (WAND; WEBER, 1990) and (MISKE *et al.*, 2014). We want to explore the connection between ontological theories and modeling languages in design tasks.

1.3 Research Objectives

The general objective of this dissertation is ***to define an approach based on Ontological and Information Visualization theories for designing concrete***

syntaxes of Domain-Specific Visual Modeling Languages (DSVMLs) to improve the DSVMLs communication capacity.

This general objective is organized into the following specific objectives:

- *To define an approach for designing concrete syntax of DSMLs*[1] *applying Information Visualization theories in the format of information visualization guidelines*;
- *To extend the approach previously defined to consider ontological theories, introducing ontological guidelines*;
- *To apply the approaches developed in the design of DSML concrete syntaxes through empirical studies*.

1.4 Research Limitations

Considering the size and complexity of the DSML treatment, its design and the theories we selected as basis, we decided to establish some limits in our proposal. These limits can be expanded in future works. At this point, we selected them as a manner to delimitate the size and complexity of our proposal, turning it more feasible.

The limits we established are:

- A modeling language can be domain-independent or domain-specific. We are dealing mainly with domain-specific languages, due to the increasing attention that the conceptual modeling community are giving to this type of modeling language;
- A modeling language can have structural and processual perspectives. In this working we are focusing on the structural perspective, as this is the focus of the UFO-based guidelines we adopted initially;
- Only concrete syntax issues are under consideration, even if the abstract syntax of a DSML can be affected when designing the correspondent concrete syntax;

---

[1] DSVMLs are commonly identified as simply DSML (Domain-Specific Modeling Language). So, in this work, we will apply the term DSML.

- A model/diagram built using a DSML can be of type elements (entities and relations) or of instance elements (instantiations of entities and relations). Our proposal is focusing on instance elements diagram application. Probably, the differences between both diagrams (type and instance elements) are smaller. However, we are not exploring this possibility here;
- We planned to evaluate our proposal through empirical studies. Studies focusing on DSMLs can have as tasks modeling language development and modeling language use. In the case of language use, it can be divided in writing and reading tasks. Our exploratory empirical studies had development and use of modeling language tasks. Our evaluation empirical studies focused on development tasks, collecting the impression of language designers as a first return of the proposal;
- Concerning PoN principles – We focused on some PoN principles, as highlighted in Chapter 4;
- Concerning UFO-based guidelines – We focused on some UFO concepts, as highlighted in Chapter 5.

1.5 Research Method

The research method adopted in this work follows the Design Science research paradigm applied in Information Systems Research (HEVNER *et al.*, 2004). Design Science methods are used to build and evaluate novel and innovative artifacts (such as new models, methods or systems) as the outcome of a research process. They are characterised by an emphasis on the construction of the artifact and the demonstration of its utility (RECKER, 2012).

The evolution of the research presented here can be viewed as a ***Design Science project*** with different iterations, each one producing its own artifact or a release of an artifact developed in a previous iteration. The ***design problem*** faced is the design of DSML concrete syntaxes. The ***artifacts*** were improvements upon two existing solutions to the problem: the PoN theory (MOODY, 2009) and the ontological guidelines for designing DSML concrete syntaxes defined in (GUIZZARDI, 2013). The artifacts are denominated PoN-S and PoNTO-S. ***Physics of Notations Systematized (PoN-S***) is a design process that applies PoN principles as information visualization guidelines. ***Physics of Notations Ontologized and Systematized (PoNTO-S)*** is an extension of PoN-S that includes ontolological guidelines to some design tasks. These artifacts can be classified as ***design theories***, considering that PoN-S and PoNTO-S are design processes for creating DSML concrete syntaxes.

To aid conducting a ***Design Science Research (DSR)***, Hevner *et al.* (2004) present a set of guidelines for conducting a DSR. We adopted these guidelines, as exposed in Table 1. Also, we organized our research as suggested in the DSR framework exposed in (RECKER, 2012), as Figure 2 shows.

Table 1: DSR guidelines and how they were considered in the current research

Guideline	Description	How the guideline was considered in the research
Design as an artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.	We develop two design processes (methods) - PoN-S and PoNTO-S.
Problem relevance	The objective of design science research is to develop technology-based solutions to important and relevant business problems.	As pointed out in Sections 1.1 and 2.1, the design of DSML concrete syntaxes is relevant to Conceptual Modeling Languages. Conceptual models are not being applied as they should. We intend to help in this sense.
Design evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.	We performed several empirical studies, as discussed in Chapter 3, and, mainly, in Chapters 4 and Chapter 5.
Research contributions	Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.	The main contributions are PoN-S and PoNTO-S approaches. The contributions we achieved are better discussed in Section 6.2.
Research rigor	Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.	We adopted a DSR, as described in the current section in order to attend this guideline. Even more, we selected as basis of the work solid theories of Information Visualization and Ontological fields, besides rely on empirical studies. More details are described in the sub-section Rigor Cycle.
Design as a search process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.	We performed an exhaustive literature review along the research. Also, we executed empirical studies (before and after each developed artifact). Besides that, intense discussions among the people involved in this research were conducted.

Communication of research	Design science research must be presented effectively to both technology-oriented and management-oriented audiences.	The list of publications we worked till the present moment is presented in Section 1.6. New ones are planned.

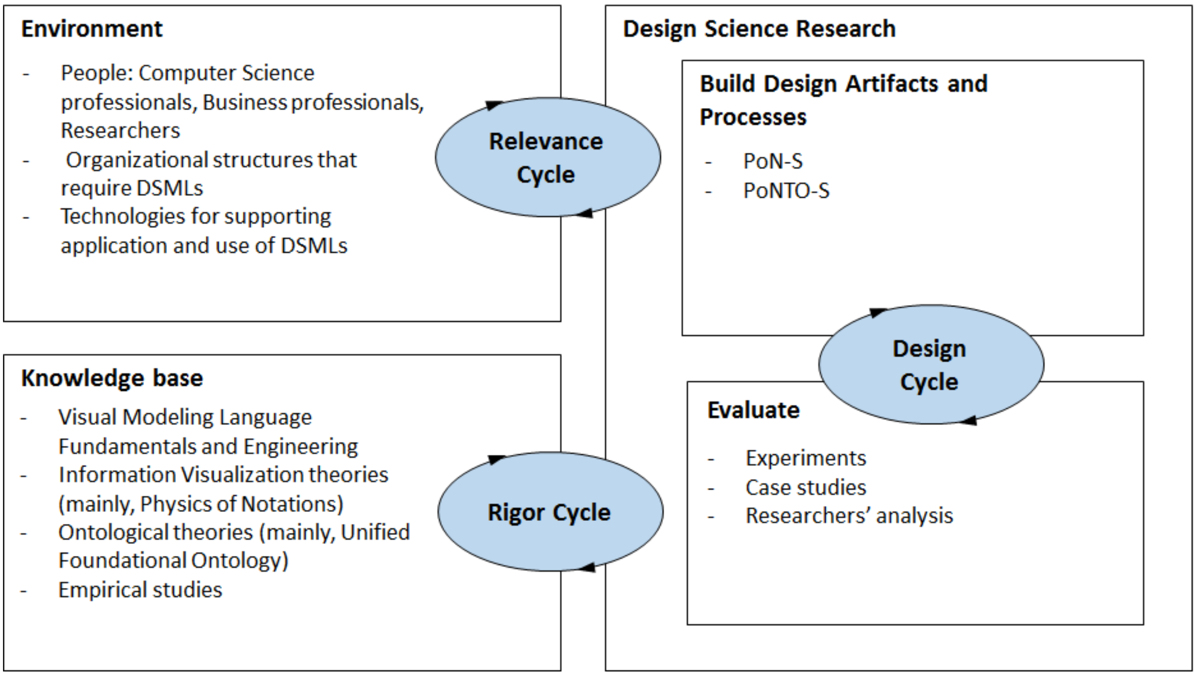


Figure 2: Adopted DSR framework. Based on (RECKER, 2012, p. 107)

Relevance cycle

The relevance cycle refers to the motivation for designing the artifacts, and how it is connected to the environment. By means of literature review, we perceived that several publications have demonstrated that Visual Modeling Languages (VMLs[2]) are a good communication tool between stakeholders of a project. However, these VMLs are not being used as they are supposed to be. Several works point out the need to invest more effort in improving the quality of both abstract and concrete syntaxes (MORRIS; SPANOUDAKIS, 2001) (GENON *et al.*, 2010) (BOBKOWSKA, 2013). This motivated us to work in improving the VML design process, focusing on DSVML, as a means to improve the quality of the resulting languages and contribute to the dissemination and use of VMLs by their probable community of users.

[2] When we apply the term Visual Modeling Language (VML) we are not differentiating between Domain-Specific and Domain-Independent visual modeling languages.

The empirical studies (exploratory and evaluative) applied were dependent on the environment, as the environment provides us with the participants, the problem domains in which build the studies tasks and also the technologies necessary to perform them. Besides this, the applied empirical studies are a useful manner to check the influence of the artifacts in the environment identified.

Rigor cycle

The rigor cycle refers to the use of a knowledge base. As identified when describing the content of this research, and detailed afterwards in Chapter 2, both the theories chosen to compose the basis for our artifacts, UFO and PoN, are solid, high-quality and disseminated theories that have been successfully applied in several cases.

Empirical studies collected empirical evidences to support the artifacts elaborated, in order to analyze the real use of these artifacts besides improve them. We applied two types of studies: experiments and case studies. Experiments were of two types: (i) Exploratory, to collect earlier evidences, before the development of the artifacts; (ii) Evaluative, to analize the use of the produced artifacts and to identify possible improvements. The case studies were applied also as an evaluation tool.

We updated the knowledge base, by proposing extensions to the PoN design process and to the UFO-based ontological guidelines. When dealing with case studies to evaluate the artifacts, the result of these activities are also new artifacts, in this case, DSMLs, built with the aid of our focus artifacts (the approaches).

Design cycle

The design cycle describes the core activities performed to elaborate and evaluate the proposed artifacts. After some initial theoretical studies, which aimed to generate a basic body of knowledge in the fields of Conceptual Modeling Engineering, Ontological theories, Information Visualization theories and Empirical Studies, we started the research by performing some experiments. These exploratory empirical studies intended to provide us insights that would guide more in-depth theoretical studies, from which we would elaborate our proposals. Also, it could guide other empirical studies that would aim to prove or refute the formulated proposals.

Our main theoretical basis for empirical studies were (JURISTO; MORENO, 2001) and (WOHLIN *et al.*, 2012). Besides, as since the beginning we had identified UFO as the ontological theory to be adopted, the empirical studies initiated with OntoUML, an ontologically well-founded profile for UML 2.0 class diagrams (GUIZZARDI, 2005a).

The first experiment aimed to analyze the ideas presented by Guizzardi (2013) on the use of ontological theories for the development of DSMLs applying these ideas on both the abstract and the concrete syntaxes of a DSML. It had as task the language use, through reading activities of instance elements diagrams. The result of such experiment is partially published in (SILVA TEIXEIRA *et al.*, 2013) and it is described in Chapter 3 This experiment and its conclusions were an important basis from which we constructed this research.

A second experiment was carried out and is also partially published in (SILVA TEIXEIRA *et al.*, 2014). The study put together alternative representations for OntoUML relations. The intention was to verify the participants behavior due to a varied representation for relations and how they would react when interpreting (so, a reading task) type elements diagrams presenting such variation. The main purpose was to prove the importance of `relator` and `role` constructs (GUIZZARDI, 2005a) and the central role that relations have in conceptual models (GUIZZARDI; WAGNER, 2008a). This experiment provided us with the idea that offering alternatives representations in the design process when suggesting representations of relations (for example, more than one manner to represent a same situation) could be useful. Also, it highlighted the importance `roles` and `relator` can have when applied in diagrams. The impact of this experiment in our research is smaller than experiments one and three. However, it had an impact in the elaboration of our ontological guidelines. More details are described in Chapter 3.

Finally, a third experiment closed the cycle of exploratory empirical studies. This experiment adopted a different background than the others, focusing on Information Visualization theories, more specifically, the Physics of Notations (PoN), which stands out as the most widely used theory for analysis and design of VML concrete syntaxes. The experiment goal was to collect impressions of language designers when using PoN to develop their tasks, so it was a language development activity. This experiment is also detailed in Chapter 3. This experiment and its conclusions were an important basis from which we constructed this research.

The literature review and the conducted empirical studies led us to the following decisions: (i) The Information Visualization theory to be used is PoN, complemented by the ontological guidelines based on UFO proposed by Guizzardi. We identified not only that these two theories are important and high-quality works, but that they also have gaps. For example, both of them do not present a design process. So, VML designers can benefit of their extensions. Since these theories have their own limitations and they are complex in themselves, we decided to deal with them at different moments, performing iterations that given rise to different artifacts. Even more, as ontological theories are not the basis to all DSMLs, it could be interesting to develop a design process based on PoN and not based on ontological theories, and finally, to develop an extension of this design process based on UFO ontological guidelines; (ii) Due to the complexity of the field we should establish some limitations that could be expanded in future researches. The limitations we defined are described in Section 1.4.

After the described activities, we were ready to build our first artifact. The following iterations were followed:

- Iteration 1: Development of PoN-S release 1, followed by an empirical study. Published in (SILVA TEIXEIRA *et al.*, 2016). PoN-S is presented in Chapter 4;
- Iteration 2: Integration of PoN-S release 2 to the ontological guidelines proposed in (GUIZZARDI, 2013), giving rise to PoNTO-S pre-release, followed by an empirical study. This empirical study is presented in Section 5.2.1;
- Iteration 3: Development of PoNTO-S release 1, adding new ontological guidelines to the initial approach defined in Iteration 2 and minor changes in the PoN-S design process, giving rise to PoN-S release 3. Also, an empirical study was applied to evaluate the approach. PoNTO-S and the aforementioned study are presented in Chapter 5.

The PoN-S and PoNTO-S approaches, generated by the iterations previously identified, were evaluated basically through empirical studies (experiments and case studies) and discussion among the researchers. As their resulting product is a DSML, more specifically, the DSML concrete syntax, we can apply two types of tasks in the studies: (i) DSML design – to have as participants of the studies language designers; (ii) DSML use – to have as participants of the studies language users performing writing or reading of models built with the DSML designed. In this work, we invested in the performance of task type (i), collecting evidences on the application of PoN-S or PoNTO-S and improving the approaches as a result of these studies[3].

1.6 Publications

Some results of this work have already been published, as listed below.

- SILVA TEIXEIRA, M. DAS G. da; QUIRINO, G. K.; GAILLY, F.; FALBO, R. DE A.; GUIZZARDI, G.; PERINI, M. P. **PoN-S: A Systematic Approach for Applying the Physics of Notation (PoN)** (R. Schmidt *et al.*, Eds.) 21st International Conference in Exploring Modelling Methods for Systems Analysis and Design (EMMSAD'16), Held at CAiSE 2016, Ljubljana, Slovenia, Proceedings. Springer International Publishing, 2016;
- SILVA TEIXEIRA, M. DAS G. da; FALBO, R. DE A.; GUIZZARDI, G. **Analyzing the Behavior of Modelers in Interpreting Relationships in Conceptual Models: An Empirical Study.** 3rd International Workshop on Ontologies and Conceptual Modeling (Onto.Com 2014)/6th International Workshop on Ontology-Driven Information Systems (ODISE 2014). Rio de Janeiro: 2014;
- SILVA TEIXEIRA, M. DAS G. da; FALBO, R. DE A.; GUIZZARDI, G. **Can Ontologies Systematically Help in the Design of Domain-Specific Visual Languages?** (R. Meersman *et al.*, Eds.) 12th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2013) - OTM Conferences. Lecture Notes in Computer Science. Springer, 2013.

[33] We also consider important to evaluate the result of PoN-S and PoNTO-S application, that is, the DSVML, in the viewpoint of language users. This type of evaluation is planned to be applied posteriorly (see Section 6.4).

1.7 Organization of this Dissertation

The remainder of this text is organized as follows:

- **Chapter 2. Theoretical Background** – This chapter describes an overview of the state of the art of the subjects we based our work on. It is divided in three main parts: Conceptual Modeling Language Engineering (as we are proposing an approach for this field), Physics of Notations (one of the theories we adopt in our approach), and the relation between ontological semantics and concrete syntax of VMLs, using UFO, another theory we adopt in our approach;
- **Chapter 3. Exploratory Empirical Studies** – This chapter presents three experiments we performed to collect earlier evidence for developing our research. The first experiment compared the quality of interpretation tasks of instance level diagrams developed using different notations. The second experiment evaluated the quality of interpretation tasks of relations represented in different ways on type level diagrams. The third experiment analyzed the results of DSML design applying PoN;
- **Chapter 4. Physics of Notations Systematized (PoN-S)** – This chapter presents PoN-S, identifying its design questions, groups of PoN principles and the design process. A case study applying PoN-S is also detailed. The approach is a result of the performed theoretical studies involving PoN (Chapter 2) as well as the evidences we collected in the exploratory empirical studies applied (Chapter 3) regarding PoN;
- **Chapter 5. Physics of Notations Ontologized and Systematized (PoNTO-S)** – This chapter presents PoNTO-S, including new ontological guidelines and identifying how the ontological guidelines are attached to the design process of PoN-S. An experiment analyzing the pre-release of PoNTO-S and a case study applying the current release of PoNTO-S are presented too. PoNTO-S is an expansion of PoN-S and it takes into consideration the performed theoretical studies regarding ontological guidelines (Chapter 2), as well as the exploratory empirical studies involving ontological guidelines (Chapter 3);

- **Chapter 6. Final Considerations** – This chapter presents an overview of the developed research, as well as the research contributions and possible influence in other researches. A brief discussion of our recommendation on which approach to apply (PoN-S or PoNTO-S) is included. Also, some limitations and future perspectives regarding PoN-S and PoNTO-S are discussed;
- **Appendix** – To complement the main text, we add complementary information: (i) We presented a draft of a new ontological guideline to deal with `collective` construct, not evaluated yet (Appendix A); (ii) We gave detailed instructions for language designers applying PoN-S and PoNTO-S (Appendix B); (iii) We described details about the OntoUML constructs chosen to be part of the first release of PoNTO-S (Appendix C).

Chapter 2. Theoretical Background

"The limits of my language mean the limits of my world."
(Ludwig Wittgenstein)

This chapter presents the theoretical foundation involved in this research. Section 2.1 presents briefly Conceptual Modeling, highlighting our main interests (fundamentals, quality and engineering). Section 2.2 presents the PoN theory. Section 2.3 presents the ontological guidelines for designing DSVML defined based on UFO. Finally, Section 2.4 presents limitations that we identified in PoN and UFO-based ontological guidelines, as well as lessons learned throughout the literature review and from which we identify gaps to be fulfilled by this work.

2.1 Conceptual Modeling Languages

One of the major purposes of conceptual modeling is to facilitate communication between users and analysts. Analysts construct a (typically diagrammatic) conceptual model to document their understanding. The model allows users to verify whether or not analyst interpretations reflect reality as perceived by the users (PARSONS; COLE, 2005).

Krogstie (2009) points out that the use of modeling as a technique for developing information systems has received more attention in recent years. Academia and Industry have demonstrated the need to better understand their problems and processes and that this can be accomplished through conceptual models. However, there is still resistance in this sense. Aranda *et al.* (2007) claim that we have at our disposal a wide range of notations to represent almost any type of information. Even so, the adoption of Conceptual Modeling in real projects is less than expected. They point out that one cause for such a problem is the lack of attention that language designers give to how languages enable effective communication among their users.

Primarily, conceptual models are used for communication between different stakeholders in some business or in the early phases of the software development process. Conceptual models support different tasks that may be related to the development or interpretation of models.

As Mylopoulos (1992) points out, conceptual models represent aspects of the physical and social world and support communication, problem solving, and negotiation of meaning among humans. Conceptual models are designed to be used by humans, not machines. Thus, it is essential that the conceptual modeling languages favor the common understanding among the stakeholders involved in this process.

A modeling language defines the modeling elements that are significant to a given conceptualization of reality and that are used to represent abstractions of this conceptualization through models. As Figure 3 illustrates, the elements of a ***domain conceptualization*** are used to articulate abstractions of certain state of affairs in reality, said a ***domain abstraction***. Such concepts exist only in the mind of the user or a community of users of a language. To document, analyze and communicate them, these concepts must be presented through an artifact, namely a ***model***. In order to represent a model, a ***modeling language*** must be defined (GUIZZARDI, 2013).

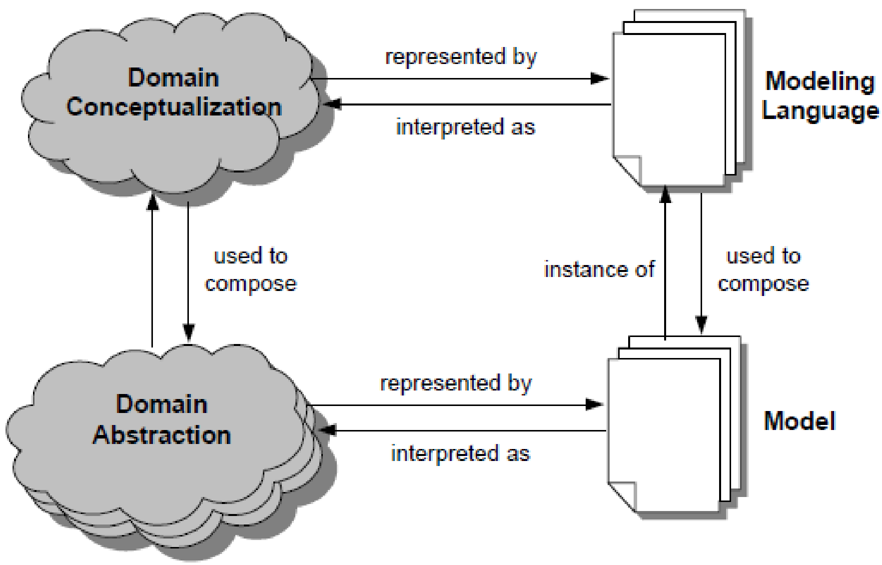


Figure 3: Relation between conceptualization, abstraction, model and modeling language.
Source: (GUIZZARDI, 2013, p. 2)

Usually a conceptual modeling language is graphical, hence the term ***Visual Modeling Language (VML)*** is applied, although we often speak simply of Modeling Language (ML). If we analyze rigorously the term, then there is a small flaw in its use:

visual is anything that is seen - use of sight sense. In this way, both textual and graphical languages are visual, since we use our vision to work with them. However, we emphasize that the term VML is used with the meaning of ***Graphical Modeling Language (GML)***. It is worth mentioning that some authors are aware of this difference and they prefer using the more precise term (GML), among them we can refer to Karagiannis and Kuhn (2002) and Schalles (2013)[4].

Figure 4 presents the modeling components defined in (KARAGIANNIS; KUHN, 2002) and (SCHALLES, 2013). Syntax describes the elements and rules for modeling and it is defined by a grammar. The grammar can be defined in the form of graphs or metamodels. Semantics describes the meaning of the language and comprises a semantic domain (or semantic schema) and a semantic mapping. The semantic domain is the description of the real-world meaning and it can be represented, for example, by an ontology. Semantic mapping connects the constructs of the syntax with their meaning in the semantic domain. Finally, notation describes the visualization of the modeling language and also takes the semantics into account.

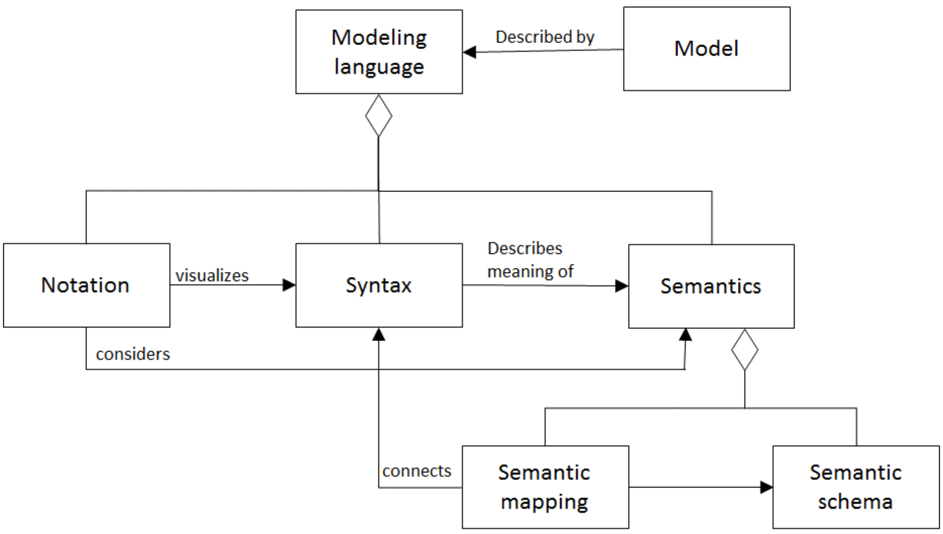


Figure 4: Modeling components. Excerpt of (KARAGIANNIS; KUHN, 2002, p. 3) and (SCHALLES, 2013, p. 15)

We should highlight that these authors use the term syntax instead of abstract syntax (a more common term). Also, they refer to notation, rather than concrete syntax. Moody, in turn, applies the term concrete syntax and he differentiates it from

[4] In this research, we decided to use the term VML instead of GML, due to its spreading.

notation. Moreover, Moody connects semantics to the concrete syntax, as follows: “A visual notation (or visual language, graphical notation, diagramming notation) consists of a set of graphical symbols (visual vocabulary), a set of compositional rules (visual grammar) and definitions of the meaning of each symbol (visual semantics). The visual vocabulary and visual grammar together form the visual (or concrete) syntax. Graphical symbols are used to symbolize (perceptually represent) semantic constructs, typically defined by a metamodel. The meanings of graphical symbols are defined by mapping them to the constructs they represent. A valid expression in a visual notation is called a visual sentence or diagram. Diagrams are composed of symbol instances (tokens), arranged according to the rules of the visual grammar” (MOODY, 2009). As Moody work is one of the basis of our research, his terminology is the one we adopted.

Other terms found in the literature are: ***Primary notation*** and ***secondary notation***. The oldest reference we find is in (PETRE, 1995), a work which suggests using of graphical representations in programming languages - which, despite the differences, are a type of modeling language. The author uses the term secondary notation, but he does not refer to the term primary notation. However, he applied the term formal notation that seems to be synonym of primary notation. Petre states that secondary notation refers to analog mappings in models and it is used to emphasize valuable layout cues that are not normally defined in the formal notation - they are secondary to language definition, but they can be used to emphasize relationships and structures that might otherwise not be accessible. What he calls analog complements are complementary information that we can visualize, such as positioning related entities close to each other. Petre points out that secondary notations may be crucial to understand the code, or in our case, the model.

Moody (2009) defines primary notation as the formal definition of a visual notation – the set of graphical symbols and their prescribed meanings. Secondary notation is about using visual variables not formally specified in the notation to reinforce or clarify meaning, for example, use of color to highlight some information. The author emphasizes that secondary notation should not be treated in a trivial manner. Moody pointed out Petre as responsible for identifying that using secondary notation is the feature that most distinguishes novice and experienced users in a notation. Also, Moody introduces the term ***visual noise*** as being an accidental use of secondary notation that produces unintended results and that causes conflicts or distorts the intended message.

Another element of the concrete syntax characterized by Moody (2009) is ***dialect***. A dialect can be understood as variants of the symbol set that compose the concrete syntax. There are several reasons to create different dialects to represent the same abstract syntax, from which we highlight the following: to attend to different user profiles and to perform different modeling tasks.

The concrete syntax defines the representational elements, whilst the ***representation strategies of models*** focus on how to organize such elements, for example through modularization, abstraction levels and layout.

Another aspect of visual representations is the use of alternative representations. According to (ROSA *et al.*, 2011), alternative representations are a resource to exhibit concepts without using the primary notation defined. Adopting this type of strategy would allow us to reach a significant level of flexibility in the language. It turns possible to define alternatives for the concrete syntax as a whole, thus obtaining different dialects, or smaller variations within the same dialect, through alternatives of some representational element. For example, to have representational elements to represent a same relation type in more than one manner, that is, alternative manners.

It is important to differentiate between ***model*** and ***diagram***. Model refers to a set of modeling elements, defined by the abstract syntax; Diagram, in turn, is a set of representational elements, defined by the concrete syntax and guided by the representation strategies of model. Authors such as Baar (2008) and Elaasar and Labiche (2011) make such distinction. This reinforces the idea that we can have different dialects and alternative representations for a same model - given by different diagrams adapting the model to different needs related to the problem domain, the modeling task, or the stakeholder profile.

Regarding to the distinction between model (related to abstract syntax) and diagram (related to concrete syntax), Moody (2009) states that visual notation issues are separated by levels. According to him, a message (remembering that the main purpose of a conceptual model is to transmit a message (communication goal)) has two basic elements: form and content. The content is related to the message meaning, and the form is referring to the message representation. Regarding the form, two levels should be considered: (i) ***Type (language) level***, which refers to the definition of the visual notation and the symbols that compose it; and (ii) ***Instance (sentence) level***, which is related to the use of the visual notation through the construction of diagrams. For example, at instance level, diagram layout issues matter. Figure 5 shows the connection between the concrete syntax levels and the language semantics connected to the abstract syntax.

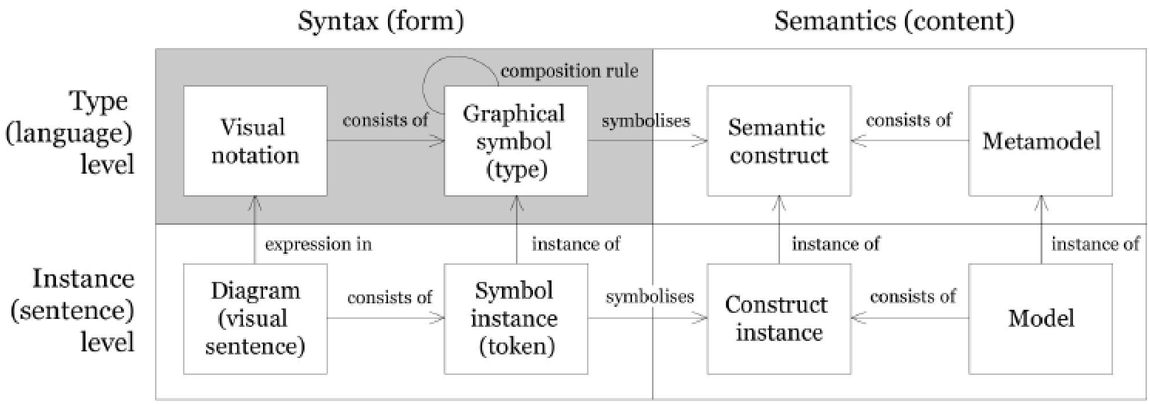


Figure 5: Concrete syntax levels and their connection to the semantic level. Source: (MOODY, 2009, p. 757)

A model (and the correspondent diagram) can be about type elements (entities and relations) or instance elements (instantiations of the type elements)[5]. In Chapter 3, we have examples of this situation. For example, in the experiment described in Section 3.1, the adopted generic notation is based on UML notation (the language level). The diagram (sentence level) of Figure 16 depicts type elements (which are considered the metamodel elements of a proposed DSML). Figure 18 is an instance elements diagram using the proposed DSML, that is, it is instantiating the type elements previously defined on Figure 16.

With respect to the scope, on one hand we have generic modeling languages, also known as ***Domain Independent Modeling Languages (DIML)***. Ideally, models created through these languages can be mapped from any problem domain. Some

[5] As the terms type and instance can be applied to refer both to levels of a language and to elements of a model/diagram, in order to reduce the confusion when applying these terms in the text we decided: (i) To use the terms language and sentence when making reference to the levels of a language; (ii) To use the terms type and instance when making reference to the elements of a model/diagram.

examples include: Entity-Relationship Model (ERM) and the Unified Modeling Language (UML). On the other hand, we have the ***Domain Specific Modeling Languages (DSML)***, which focus in specific domains and so they have their application restricted[6] to these domains. An example is the Manufacturing Execution Systems Modeling Language (MES-ML) (WEISSENBERGE; VOGEL-HEUSER, 2012).

Recently, there has been a growing interest in DSMLs and the support they provide for domain understanding and communication (TOLVANEN *et al.*, 2004) (GUIZZARDI, 2013). DSMLs provide significant support to users of conceptual models, which are often unfamiliar with the syntax and semantics of a conceptual modeling language.

With respect to the perspective, modeling languages can be classified in (ROEBUCK, 2011): (i) ***Structural***, which focuses on the static characteristics of the model, that are, its entities and relations among them. An example is the ERM; (ii) ***Procedural*** / Behavioral, which focuses on the description of dynamic characteristics. An example is the BPMN language. Languages can adopt more than one perspective, as it is the case of UML.

Regarding the use of models, two basic types of tasks are usually performed: ***Coding*** and ***decoding***, also called writing and reading, or development and interpretation. Some researchers have considered that different user capabilities are required according to the type of task performed. This is the case of Parsons and Cole (2005) that presented guidelines for the empirical evaluation of diagrammatic conceptual modeling techniques separated by task – "Options for evaluation include: assessing the ability of developers to construct models that capture requirements ('write' tasks), and assessing the ability of readers of models to extract information contained in them ('read' tasks)" (PARSONS; COLE, 2005).

Mayer (apud SCHALLES, 2013) considers the following components as fundamental to the explaining information by humans process, as it is the case of conceptual models: (i) The problem domain (content) to be modeled or interpreted; (ii) The characteristics of the modeling language to be used for modeling or

[6] This restrinction allows the designer to create more focused languages to a certain domain, dealing with the specific characteristics of the domain.

interpreting the problem domain; (iii) The individual characteristics of the model reader or developer. Also, we believe in the relation between these aspects, arguing that a modeling language should be the best reflection of the problem domain and should be adjusted to the stakeholders profile and modeling task at hand.

Quality of Concrete Syntax of Conceptual Modeling Languages

The notion of quality in CML is still immature, and there are only some identified properties of what would be a “good” conceptual representation (NELSON *et al*., 2012). Among these properties, we highlight the following, which are related somehow to the design of VMLs[7]:

- ***Appropriateness*** - It is a subjective concept, as someone must judge how appropriate something is, and this judgment can vary. In some situation, it could be possible to formally define and establish measures for appropriateness. In other cases, even if a formal definition is identified, the use of measures would be dependent to a human judgment.

  The SEQUAL Quality framework (KROGSTIE, 2009) identifies six quality areas for modeling languages, all of which involve appropriateness. Such areas highlight interpretation tasks, leaving aside development tasks. Anyway, a model to be properly interpreted should at first be well-defined. Also, such approach does not emphasize the role of concrete syntax. It is implicitly contained in some areas, such as comprehensibility appropriateness;

- ***Comprehensibility / Understanding*** - To achieve success in communication, the model must be properly read, processed and assimilated. Aranda *et al.* (2007) use such argument when claiming that comprehensibility is an essential quality of communication artifacts.

  Aranda *et al.* (2007) listed some difficulties for evaluating comprehensibility, namely: (i) Equivalence of information - if we are going to use some reference model, we must be careful not to make a very simplistic or irregular

---

[7] The quality properties described here are some terms commonly found in several works that deal with quality criteria of modeling languages. They can be analized separately or in tandem when designing a modeling language.

comparison; (ii) Accessibility to participants - finding the appropriate participants to the profile required by the study is difficult. Besides that, we have to consider the sample size; (iii) Researcher tendency - if the researcher is testing something that s/he has developed, some care must be taken not to be biased in planning and even analyzing the study results;

- ***Effectiveness*** - Gurr (1999) and Moody (2009) defend the need of obtaining efficacy in visually represented models.

  Gurr (1999) highlights as quality criterion for diagrammatic representation the effectiveness and he makes syntactic, semantic and pragmatic considerations in the attempt to achieve adequate effectiveness.

  PoN (MOODY, 2009) presents principles that can influence visual notation. PoN main quality criterion is cognitive effectiveness. Moody states that cognitive effectiveness influences the ability to communicate visual notations to different stakeholders involved with the model;

- ***Usability*** - Bobkowska (2005a) (2013) states that usability issues are centered on concepts such as easy to learn, easy to use, productivity and user satisfaction. Usability techniques can avoid annoying, artificial or complicated features that make it difficult to perform tasks or to induce errors.

  Schalles (2013) claims that complexity and visual differentiation defined in the language metamodel influence the usability of this. He presents a method for evaluating usability in Graphical Modeling Languages. According to Schalles, the following attributes can be used to measure the usability of GML: (i) Ability to learn; (ii) Ability to memorize; (iii) Effectiveness; (iv) Efficiency; (v) User satisfaction; and (vi) Ability of visual perception (visual perceptibility).

Complementing these properties, we claim that in researches of evaluation and design of modeling languages, focusing on the visual aspect, the importance of the stakeholder involvement and application of empirical studies are significant, as in (GENON *et al*, 2012) and (CHO *et al.*, 2012). So, they should be considered when evaluating the resulting modeling language, besides assessing some quality properties.

2.1.1 Design of Conceptual Modeling Languages

As an artifact, ideally, a CML should be developed through a design process. This process must address the abstract and concrete syntaxes.

During the design process, the language designer should take decisions that can affect both language and sentence levels - what will be done at language level and what can be done at sentence level. Language level decisions should be addressed by the language designer and typically are mandatory. Although modelers are responsible for sentence level decisions, these are guided by what is established in the language level, so it is the language designer that would establish what is possible to the modeler to do.

There are several works in the literature related to the desing of CMLs. Wouters (2013) presents a procedure to perform the mapping between the abstract and concrete syntaxes, initiating by the concrete syntax. Fish and Storrle (2007) provide guidelines to increase language efficiency, organized in categories, such as mapping between the concrete and abstract syntaxes, and mapping between abstract syntax and semantics.

Highlighting the influence that end-users should have in the design of a DSML, Cho, Gray and Syriani (2012) propose a framework for developing DSMLs focused on end-user action, called Modeling Language Creation By Demonstration (MLCBD). They present an iterative and incremental approach for building a DSML, illustrated in Figure 6. In their approach, the DSML design begins by eliciting the DSML requirements, when domain experts describe what capability is required and how that is represented in their domain. Based on the requirements, the language designer should identify the concrete syntax, design the abstract syntax, and specify semantics. Finally, domain experts should verify the DSML and give formative feedback that may lead to an iteration requiring more development (CHO *et al.*, 2012). MLCBD has as input a set of domain model examples provided by an end-user. From these examples, the concrete syntax is extracted and from the representational elements, the abstract syntax is induced. The differential in this approach is the use of examples to infer the concrete syntax, which is generated before the abstract syntax. In this design process, the set of representational elements is identified and taking them as basis the set of modeling elements is established.

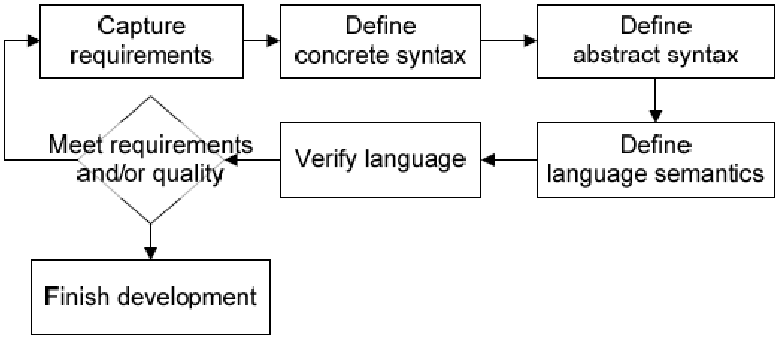


Figure 6: DSML development process. Source: (CHO *et al.*, 2012, p. 22)

2.2 The Physics of Notations (PoN) Theory

One of the most widespread work in the area of visual aspects of modeling languages, is the **Physics of Notations (PoN) Theory** (MOODY, 2009). PoN defines a theory for designing cognitively effective visual notations, that is, notations that are optimised for processing by the human mind. It consists of nine principles based on theories and empirical evidence from a wide range of fields, such as Communication, Semiotics, Graphic Design, Visual Perception, Psychophysics, Cognitive Psychology, Human-Computer Interation (HCI), Information Visualization, Information Systems, Education, Cartography, and Diagrammatic Reasoning (MOODY, 2009).

PoN can be used to compare, evolve, improve and build visual notations. The approach adopts as core quality criterion **cognitive effectiveness**, defined as the speed, ease, and accuracy with which a representation can be processed by the human mind (MOODY, 2009).

In PoN, the visual notation design process distinguishes three spaces (MOODY *et al.*, 2010):

- Problem space: The information content to be expressed;
- Design space: The set of all possible graphical encodings of a particular notation. There are ***visual variables*** which can be used to graphically encode information. They define the visual alphabet for constructing a visual notation. Any graphical symbol can be defined by specifying particular values for the visual variables (e.g., shape = rectangle; color = green).

In this sense, several studies, such as (MOODY, 2009), (SCHALLES, 2013), (FISH; STORRLE, 2007), (GENON *et al.*, 2010) make reference to the

Semiology of Graphics, proposed by Bertin (1983), which identifies **visual variables** (or **properties**) (see Figure 7) and their characteristics.

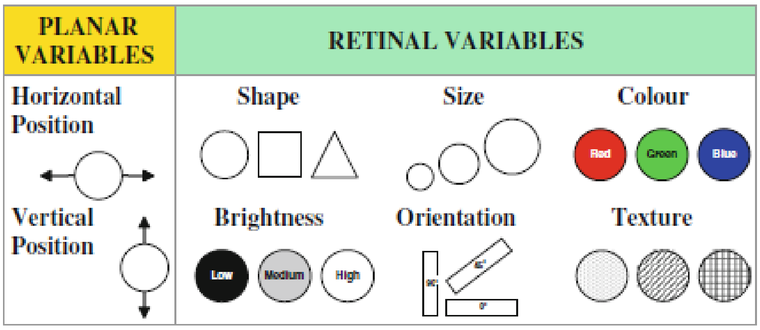


Figure 7: Visual variables. Source: Bertin Apud (MOODY, 2009, p. 761)

- Solution space: Designing cognitively effective visual notations is a problem of choosing the most cognitively effective representations for the concepts of the problem space from the infinite possibilities in the graphic design space.

PoN identifies nine principles for designing cognitively effective visual notations. The complete set of principles and the criteria associated with each of them are presented in Figure 8. Next, each principle is described.

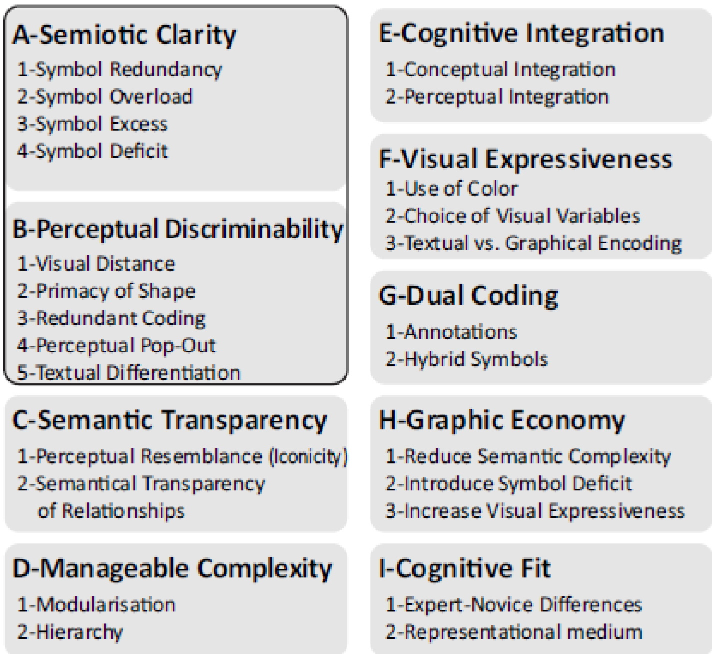


Figure 8: PoN principles and their criteria[8]. Source: (STORRLE; FISH, 2013, p. 109)

[8] Semiotic Clarity and Perceptual Discriminability principles are highlighted in the figure because these are the principles with which the work of Storrle and Fish focus on.

Semiotic Clarity

"There should be a 1:1 correspondence between a metamodel construct and a graphical symbol" (MOODY, 2009).

As Figure 9 shows, in case of no attendance of this principle, the following anomalies can occur: symbol deficit (when a semantic construct is not represented by any symbol), symbol overload (when the same symbol is used to represent multiple semantic constructs), symbol excess (when a symbol does not represent any semantic construct), symbol redundancy (when a semantic construct is represented by multiple symbols).

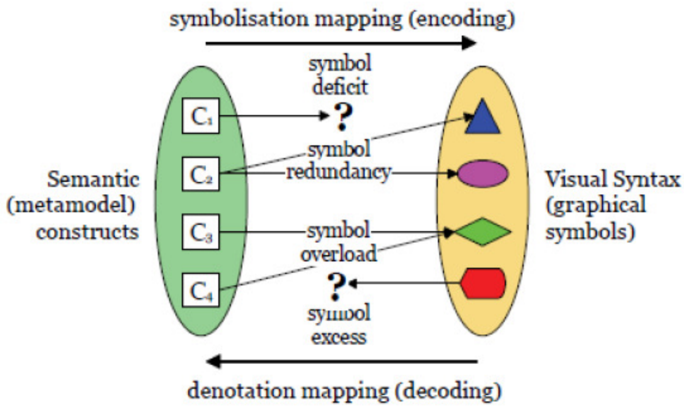


Figure 9: Anomalies identified by Semiotc Clarity principle. Source: (MOODY, 2009, p. 762)

The design goal should be maximizing expressiveness by eliminating symbol deficit; maximize precision by eliminating symbol overload; and maximize parsimony by eliminating symbol redundancy and excess.

Since these anomalies are unwanted, we should try to satisfy this principle during language design. However, to attend completely this principle and depending on the number of language constructs used, the solution can impact on other principles, such as Complexity Management and Graphic Economy. So, a designer should consider the language requirements before deciding how to implement this principle – if necessary, anomalies can remain in the language.

Perceptual Discriminability

"Symbols should be clearly distinguishable from one another" (MOODY, 2009).

The Perceptual Discriminability principle highlights the need to easily and accurately differentiate symbols. This is an important issue considering that accurate discriminations between symbols is a pre-requisite for accurate interpretation of diagrams (MOODY *et al.*, 2010).

This principle is complementary to Semiotic Clarity. While Semiotic Clarity focuses on defining the symbols needed (based on the language constructs), Perceptual Discriminability is centered on the need to differentiate between such symbols, which should ideally respect the semantic distance and properties of concepts reflected by the correspondent constructs.

Discriminability between symbols can be operationalized by ***visual distance*** between symbols, that is, the number of visual variables on which they differ and the size of these differences (MOODY, 2009). Moreover, the visual distance between symbols should be consistent with the semantic distance between the constructs they represent (MOODY *et al.*, 2010). This notion, said semantic congruence, is mainly associated to the shape visual variable. In general, similar shapes should be used to represent similar constructs (MOODY *et al.*, 2010).

Shape is the primary base to represent elements. So, using shape family to represent similar constructs could be a good strategy. An important communication technique to reduce errors is redundant coding, using more than one visual variable to identify a symbol, so increasing the facility to distinguish between symbols. The use of redundant coding reinforces the visual distance between symbols. However, we need to find the right balance, so that redundancy does not become a problem. Textual differentiation, on the other hand, is cognitively ineffective. Symbols that differ only by text have visual distance equal zero, that is, they are visually identical. Text is not a visual variable. Text is recommended to be applied when developing instance elements diagrams (MOODY *et al.*, 2010).

Semantic Transparency

“Use symbols whose appearance suggests their meaning” (MOODY, 2009).

This principle refers to the easiness with which the meaning of a symbol is recognized. It acts as a complement to Semiotic Clarity and Perceptual Discriminability. Visual notation design is not just a matter of defining a symbol for each construct (Semiotic Clarity), but to be able to distinguish between them (Perceptual Discriminability), and easily deduce the meaning of each symbol (Semantic Transparency).

Semantic Transparency can be applied by means of two types of associations between a construct and its representation:

- Semantic immediacy: it represents direct associations. They are literal, as, for example, between a stickman and a person;
- Semantic translucency: it represents indirect associations, mnemonics. They should resemble concrete objects that are somehow associated with the referent concepts (MOODY *et al.*, 2010), as for example to represent an agent as a stickman with hat and sunglasses (a reference to 007 character of James Bond movies). This type of association is not immediate, but once it occurs, it can be easily remembered.

Semantic transparency is not a binary state, but a continuum, as depicted in Figure 10.

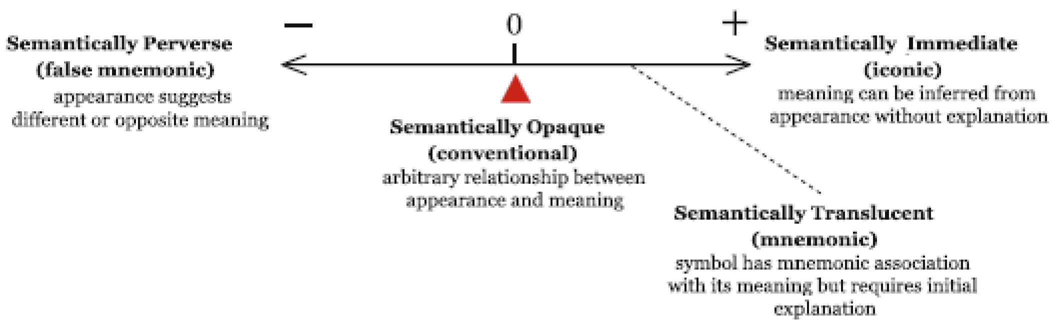


Figure 10: Semantic transparency continuum. Source: (MOODY *et al.*, 2010, p. 155)

Icons are symbols that perceptually resemble their referent concepts (MOODY *et al.*, 2010). They are especially effective for communication involving novices, because they can be directly perceived or easily learned. Abstract symbols, in turn, reduce semantic immediacy issue. However, if the symbols are properly defined, after some language learning stage, we win on simplicity matter. Moreover, abstract symbols are even important when dealing with DIMLs. Finally, small variations on a shape – generating new shapes connected – is a good choice to keep similarity between concepts.

Visual Expressiveness

“Use the full range and capacities of visual variables” (MOODY, 2009).

This principle is related to the use of visual variables and their values. Using a variety of visual variables results in a perceptually enriched representation that exploits multiple visual communication channels (MOODY *et al.*, 2010). This principle is bound to others (except Dual Coding), because it is about the expressiveness of the visual variables set we explore in a concrete syntax, and the other principles are built upon such set.

Only when a visual variable is used to encode information (it is part of the primary notation), it adds value to visual expressiveness. The use of visual variables considered as free variables (secondary notation) does not add value to the notation. For example, if a notation does not establish color as part of its concrete syntax, but in a modeling task, the modeler can apply color when developing a diagram, to highlight an entity, then color is not increasing the visual expressiveness of that language.

The choice of visual variables should not be arbitrary. It should be based on the nature of the information to be conveyed. Different visual variables have properties that make them suitable for encoding different types of information (MOODY *et al.*, 2010).

Usually, icons are more efficient than geometric abstract shapes. Colour is used to communicate rather than to decorate. It helps both to distinguish between symbols (Perceptual Discriminability) and suggest their meaning (Semantic Transparency). Ideally, colour should be used in a redundant way to reinforce the meaning of a symbol. However, if not used carefully, colour can undermine communication (MOODY *et al.*, 2010).

Graphic Economy

“The number of different graphical symbols should be cognitively manageable” (MOODY, 2009).

This principle works with the idea of keeping the number of symbols under control. This principle conflicts with others, since the other principles indicate that we should add details in the concrete syntax (each symbol with a different purpose), while Graphic Economy indicates that details should be minimized. Thus, this principle can work as a reminder that we should keep a balance between necessary and sufficient symbols in the concrete syntax.

Graphic complexity refers to the number of different symbol types used in a notation (MOODY *et al.*, 2010). When such number increases, then the understanding level decreases, especially for novice modelers. The human capacity for distinction of alternatives is around six, which should be considered an upper bound to graphic complexity.

Moody (2009) states that there are three strategies to deal with graphic complexity, namely: (i) Reduction of semantic complexity by changing the language abstract syntax and, therefore, the concrete syntax; (ii) Introduction of symbol deficit, thus affecting Semiotic Clarity principle; and (iii) Increase of visual expressiveness by increasing human capacity to discriminate between symbols rather than increasing number of symbols. Thus, we can notice that reducing graphic complexity (which is the goal of this principle) implies affecting other principles, mainly Semiotic Clarity, Visual Expressiveness and Perceptual Discriminability.

Dual Coding

"Use text to complement graphics" (MOODY, 2009).

Perceptual Discriminability and Visual Expressiveness warn against using text to encode information. This does not mean that using text should be avoided. It means that text and graphic should be applied together in a more effective way. The idea is to use text as a complement to graphic information, not as a substitute, thus acting as a form of redundant coding.

Usually, text has a complementary role at language level (metamodel). However, text has a fundamental role at the sentence level (diagrams), known as labeling. In particular, labels play an important role in interpretating diagrams and in defining their real world semantics (MOODY *et al.*, 2010). In this sense, text should be used to reinforce expressiveness, such as to describe relationship cardinalities.

Complexity Management

“Include explicit mechanisms for dealing with complexity” (MOODY, 2009).

A well-known problem with visual representations is that they do not scale well. To deal with this problem, Moody (2009) suggests the use of modularization and hierarchical structures. According to the Cognitive Load theory cited in (MOODY, 2009), when reducing the amount of information we have to process at a time, we increase the speed and accuracy of understanding. Moreover, it is not enough to define constructs at language level, because for them to be effective at sentence level, it is necessary to define some conventions for modularization process (MOODY, 2009).

This principle affects both language and sentence levels. To properly manage complexity, we need that some representational elements exist (language level) and a process to guide on how to use these elements the best way possible (sentence level).

Cognitive Integration

“Include explicit mechanisms to support integration of information from different diagrams” (MOODY, 2009).

This principle is related to the integration of information spread in several diagrams. Since different diagrams would probably be developed in the same endeavor, we should be aware of the need to integrate information spread over such diagrams. Such integration process should be both cognitive and perceptual.

In this sense, it is important to have a whole model view, that is, a view of the whole model without details, to use contextualization techniques (focus + content), so that the part of a model of current interest (focus) is displayed in the context of the model as a whole, and to use a "wayfinding" technique, guiding model readers to move within a plane in a safe and informed manner (MOODY, 2009).

This principle is close connected to Complexity Management. It is not enough to divide information in different diagrams to manage complexity. The elements spread in such diagrams should be connected easily. Also, there is a need to maintain traceability on where an information we need is. In fact, using multiple diagrams places additional cognitive demands on the reader to mentally integrate information from different diagrams, and keep track of where they are in the system of diagrams.

Cognitive Fit

“Use different visual dialects for different tasks and audiences” (MOODY, 2009).

This principle suggests that instead of having a single notation for all situations (‘one size fits all’), different representations can be set, according to modeling tasks and stakeholders profiles.

Interactions Between the PoN Principles

During the design of a visual notation, ideally, the objective should be to satisfy all principles in an acceptable level, rather than to optimize some at the expense of others (MOODY *et al.*, 2010), unless the opposite is explicitly determined by the language requirements. Anyway, we need to understand the interactions between PoN principles.

The knowledge about the interactions between the principles can be used to identify where principles conflict with one another and where principles support each other (MOODY, 2009). Figure 11 shows the interaction between PoN principles, highlighting which principle, when applied, impacts on other principles and thus, establishing a cause-effect relationship. This cause-effect relationship can be positive (increasing the affected principle), negative (reducing the affected principle) or both positive / negative (depending on the way the cause principle is applied). In this figure, red cells indicate a negative effect; green cells indicate a positive effect; orange cells indicate either a positive or negative effect, depending on the situation; grey cells indicate that there is no cause-effect relationship between two principles.

Cause	Effect								
	Semiotic Clarity	Perceptual Discriminability	Semantic Transparency	Complexity Management	Cognitive Integration	Visual Expressiveness	Dual Coging	Graphic Economy	Cognitive Fit
Semiotic Clarity								±	
Perceptual Discriminability						+			+
Semantic Transparency		+							±
Complexity Management								-	+
Cognitive Integration	-			+				-	
Visual Expressiveness		+						+	±
Dual Coding									+
Graphic Economy		+		+		-			+
Cognitive Fit									

Figure 11: Interactions between PoN principles: + (green cell) indicates a positive effect, - (red cell) indicates a negative effect, and ± (orange cell) indicates that either a positive or negative effect depending on the situation. Source (MOODY, 2009, p. 774)

Some Applications and Analysis of PoN

PoN has been applied both to analysis and design of VMLs. Most works use PoN to evaluate some existing language notation, which usually results on change suggestions, causing a redesign. Few works effectively use PoN to design a new VML. We believe that this is an indication that PoN theory needs to be clarified with regard the notation design process. Cases that result in redesign are also our interest because their evidences could help us in collecting data for a design process.

The PoN theory brought to the community some benefits, as a growing awareness of the need to apply design principles on the language concrete syntax, besides the establishment of the abstract syntax. One example is presented by (GENON *et al.*, 2010), which evaluated Use Case Map (UCM). We highlight some claims they made concerning PoN application: (1) "It is a time-consuming endeavour"; (2) "Doing such analyses requires a deep understanding of the theory and of the rationale behind its principles, and so training becomes essential."; (3) "One of the most challenging tasks is to elicit the semantic constructs of the language itself. This task is not as straightforward as claimed by the Physics of Notations […] and it is even harder when no metamodel is available"; (4) "The availability of metrics or clear criteria for evaluating some of the principles is uneven."; (5) "One needs to understand trade-offs when introducing solutions. Many of the principles represent conflicting goals, and solving an issue related to one principle may create issues with other principles along the way, hence the need to understand how and by whom the notation is intended to be used". Interpreting these points, we have that items 1 and 2 highlight the need to have a well-defined evaluation process, which will help to reduce the evaluation and learning times for implementation of the framework; Item 3 brings us to the use of ontologies and using them as a support to the real-world mapping for the metamodel and so for the visual notation; Item 4 is a reminder of the importance of finding ways to transform the results in numbers that we can work with because these allow us to compare and associate to pattern values; Item 5 reinforces one of the questions on PoN theory and also related to other theories: its principles may conflict between each other and we need to make decisions about which will be the principles that we will highlight, what will depend on our knowledge of other aspects that can influence language design such as stakeholder profile, task modeling, domain problem type.

Caire *et al.* (2013) present five studies involving the concrete syntax of i* (a modeling language in the Requirements Engineering field), one of them applying PoN theory by specialists. The research question of this work is how to design user-comprehensible visual notation. As a conclusion, they highlight the importance of the Semantic Transparency principle and the involvement of end-users in the notation design, more specifically novice modelers. Moreover, they point out difficulties to operationalize PoN. In addition, they reinforce the importance of collecting empirical evidence as a support to the VML design, in a clear enhancement of pragmatic issues for notation analysis and design.

Among the main evidences identified in (CAIRE *et al.*, 2013), we highlight: (i) Symbols generated by novice modelers are more semantically transparent than those generated by specialists, thus contradicting the general notion that professional experts are more qualified for symbol design in a notation; (ii) Using explicit design principles significantly improves semantic transparency.

The Socio-Technical System Notation (STSN) (MISKE *et al.*, 2014) is a DSML designed for the Business field. STSN incorporates ontological, semantic, and visual design improvements. Concerning ontological and semantic clarity and completeness, STSN is based on BWW Ontology (WAND; WEBER, 1990) and Moody Semantic Clarity Model (MSCM), an alternative name the authors use refering to PoN (MOODY, 2009). The visual design was informed by PoN (MISKE *et al.*, 2014). However, the publication does not present details on how the authors applied these theories. It only indicates that the language objectives, among them being easy to use and recognize, require the application of PoN principles.

In the context of CHOOSE (BOONE *et al.*, 2014), an enterprise architecture approach for small and medium-sized enterprises (SMEs), a notation was designed supported by PoN. We can clearly notice the intention to develop a language benefiting a profile: novice modelers. Regarding the application of PoN principles, Boone *et al.* (2014) highlight the following: (i) Semiotic Clarity is applied to obtain an unambiguous notation that inherently avoids misconceptions; (ii) Perceptual discriminability is very important because this notation is to be used by novices and the requirements for discriminability are higher for novices than for experts; (iii) Semantic Transparency can be achieved designing semantic transparent symbols using icons, which lead to a faster recognition and recall of the constructs. Besides, they especially enhance the comprehensibility of the notation for novice users; (iv) Visual Expressiveness is not clearly characterized in the work. It only points out use of color, a powerful mechanism that can become a problem when diagrams are printed on black-and-white style; (v) Complexity Management is considered an important PoN principle in the DSML, because novices have more difficulties dealing with complexity than experts; (vi) Dual Coding is somewhat less addressed in the language; (vii) Cognitive Integration is not incorporated in the language. When SMEs grow and more detail needs to be added to the models, it might be useful to map the CHOOSE models on the ArchiMate standard; (viii) Graphic Economy is applied through Visual Expressiveness - Manipulating multiple visual variables reduces the need to lower the amount of symbols; (ix) Cognitive Fit is not included in this research because in general most users of the target group are novices in enterprise modelling.

Regarding extensions of PoN, Storrle and Fish (2013) claim that PoN still needs expansions to be operationalized, and propose ways to operationalize such approach focusing on analysis tasks of modeling languages, establishing a series of measures that complement PoN original proposal, increasing the detail level of some PoN principles, and pointing out some gaps in such principles. However, a process for applying them is not identified. These extensions are motivated by the following reasons (STORRLE; FISH, 2013), among others:

- While PoN seems to maintain that graphemes are atomic and small in number, they are in fact structured and numerous, at least for practical languages such as UML or BPMN. Managing a significant number of constructs and following the PoN principles, even partially, is a hard work, because we need to keep in mind not only the good design of one symbol that represents one construct but all constructs set;
- PoN does not consider that a grapheme can be structured and composed. However, this is an aspect presented in different VMLs and therefore needed to be recognized as a possibility for concrete syntaxes.

2.3 On the Relation Between Ontological Semantics and Language Concrete Syntax

> "An *ontology* is an explicit specification of a conceptualization" (GRUBER, 1995)

Gurr (1999) argues that the stronger the match between the domain conceptualization and the model that represents this conceptualization, the easier it is to reason using this model. The interpretation of a diagram by a user must correspond precisely and uniquely to the problem domain being represented. A direct consequence is that the more we know about the domain being represented, the better we can systematically explore its properties to design a language. This stands out particularly when we are dealing with DSMLs. This motivated us to work with ontological theories: a well-founded reference ontology can be used as a semantic schema to produce the abstract and concrete syntaxes of a modeling language. So, we suppose that the quality (in this case, a suitable representation of the domain conceptualization) arises since the first artifact[9] of the language engineering process.

Ontologies are a means to formally model the structure of a system, that is, the relevant entities and relations that emerge from its observation, and which are useful to some purpose (GUARINO *et al.*, 2009). Guarino (1998) classifies ontologies according to their level of generality in the following categories: (i) Top-level or ***Foundational Ontologies*** describe general concepts that are independent of a particular domain or task, such as time, space, object, and event; (ii) ***Domain Ontologies*** describe the vocabulary of a specific domain (for example, food industry), specializing the terms of a foundational ontology; (iii) Task Ontologies describe a general activity (for example, diagnosis), specializing the terms of a foundational ontology; (iv) Application Ontologies identify concepts related to a given domain and task, specializing both a domain ontology and a task ontology.

Another classification is proposed by Scherp *et al.* (2011). Their proposal focuses on the structural perspective, adding a new level between foundational ontologies and domain ontologies, called ***Core Ontologies***. A core ontology structures the concepts of a larger domain (for example, Law) that is expanded in different domains (for example, Criminal Law).

In the recent years, there has been a growing interest among the conceptual modeling community to rely on ontologies (GUIZZARDI, 2005a). When models are constructed based on ontological theories to improve the theory and practice of Conceptual Modeling, this field is called ***Ontology-Driven Conceptual Modeling (ODCM)***. These ontological models can be applied in the evaluation of conceptual modeling languages and in the development of the languages (providing, for example, methodological guidelines and design patterns) (GUIZZARDI; HALPIN, 2008).

Ontologies can be the basis for designing Modeling Languages, providing the semantic schema of the languages. DIMLs can be based on foundational ontologies. This is the case of OntoUML, an ontologically well-founded profile for UML 2.0 class diagrams (GUIZZARDI, 2005a). In case of DSMLs, domain ontologies can be used,

[9] Considering a design process in which the modeling elements (abstract syntax) are defined previously to the definition of the representational elements (concrete syntax).

such as in the REA-DSL language (SONNENBERG *et al.*, 2011), based on Resources-Events-Agents (REA) ontology which is a business process modeling ontology (GEERTS; MCCARTHY, 2002).

The link between ontology and modeling languages has been studied by many researchers, most of them focusing on the VML abstract syntax. Henderson-Sellers (2011) claims that an ontology is fundamental to ensure that modeling elements have well-defined semantics. Verdonck *et al.* (2015) point out different connections between ontologies and conceptual modeling that can be explored. They claim that ontologies are quite useful to evaluate modeling languages (providing a well-defined representation of the domain conceptualization to be compared with the modeling elements), as well as can be used as a theoretical foundation for conceptual models (providing the entity and relation types that can be used to develop a clear model).

Guizzardi (2007a) explores the connections of ontologies and modeling languages to develop modeling languages of high quality. He claims that a language is appropriate to represent phenomena in a given domain if the metamodel of this language is isomorphic to the ideal ontology of that domain, and the language only has as valid specifications those whose logical models are exactly the logical models of the ideal ontology. As such, this work establishes a solid bond between ontologies and language abstract syntax.

Foundational ontologies have been explored as a way to improve the quality of conceptual modeling languages. For example, Everman and Wand (2001) report the results they obtained by mapping UML constructs to the BWW ontology. Guizzardi (2005) incorporated into the UML 2.0 metamodel some ontological distinctions and axioms defined by the Unified Foundational Ontology (UFO), giving rise to the OntoUML language.

Saghafi and Wand (2014) developed a meta-analysis of empirical research about the impact of ontological guidance (based on BWW) on the understanding of conceptual models. They conclude that an ontologically-guided model can lead to a better (clearer and more complete) representation of the real-world, resulting in improved user understanding.

However, a question remains open: can ontologies be successfully applied in the design of concrete syntax as it has been applied for abstract syntax? Guizzardi *et* *al* (GUIZZARDI; FERREIRA PIRES; VAN SINDEREN, 2002) (GUIZZARDI; FERREIRA PIRES; VAN SINDEREN, 2005) (GUIZZARDI, 2013) take a step forward discussing how ontological meta-properties can be used in the analysis and (re)design of visual DSMLs, considering both abstract and concrete syntaxes. In (GUIZZARDI, 2013), Guizzardi explicitly considers the connection between some PoN principles and a number of ontological guidelines put forth by UFO. Before discussing such guidelines, it is important to present UFO.

2.3.1 Fundamentals of the Unified Foundational Ontology (UFO)

UFO is based on different theories, such as Formal Ontology, Philosophy, Logic, Linguistics and Cognitive Psychology. UFO was created with the purpose of providing ontological foundations for conceptual modeling languages (GUIZZARDI, 2005a). Till now, UFO has been successfully applied for ontological analysis and design (focusing on abstract syntax) of several conceptual modeling languages. For an overview see (GUIZZARDI *et al.*, 2015).

UFO is organized in three parts: UFO-A, UFO-B and UFO-C. UFO-A is the core of UFO. It is an ontology of `endurants` (common objects that persist over time, such as a person or an orange), so it distinguishes among categories of object types. UFO-B is related to `perdurants` (events). UFO-C focuses on social entities.

Here, we are particularly interested in UFO-A, whose basic distinction is between `individuals` (instances) and `universals` (types). `Individuals` are entities that exist instantiating a number of `universals` and possessing a unique identity (for example, a book), so, they are entities that exist in reality, while `Universals` are standard features that can be instantiated by different `individuals` (GUIZZARDI, 2005a). Figure 12 presents a fragment of the UFO-A model.

A fundamental metaproperty used to distinguish among categories of object types on UFO is *rigidity* (and the associated notion of *anti-rigidity*). A type T is rigid iff every instance of T is necessarily an instance of T. In contrast, a type T' is anti-rigid iff for every instance x of T' there is a possible situation in which x is not an instance of T'. For example, there are the types Person and Student: instances of Person are necessarily so (Person is a rigid type); in opposition, instances of Student are merely contingently so (Student is an anti-rigid type) (GUIZZARDI, 2005a).

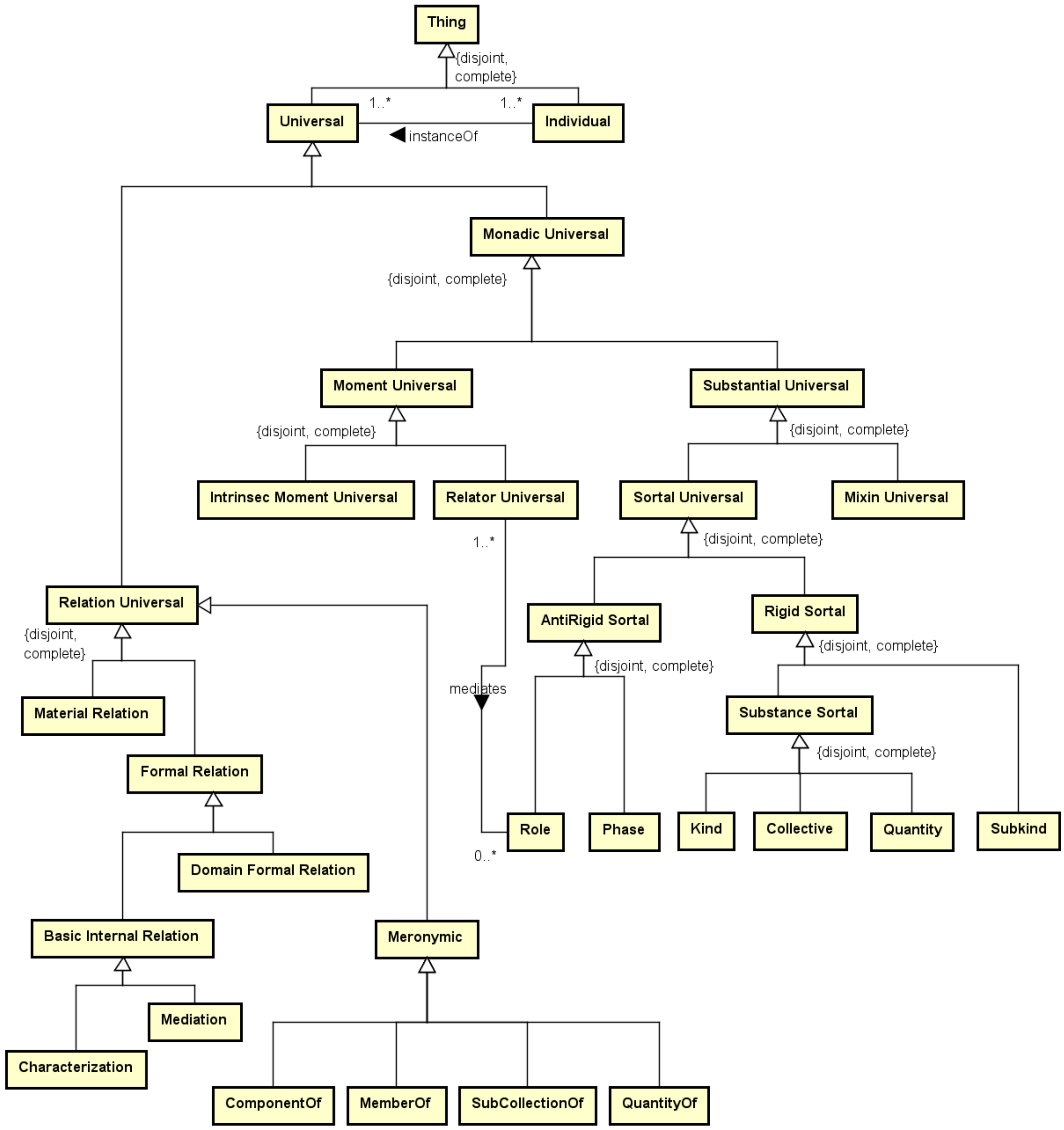


Figure 12: UFO-A structure. Source: (GUIZZARDI, 2005a)

`Kinds` are sortal rigid types that provide a uniform principle of identity for their instances (for example, Person). `Subkinds` are sortal rigid universals that carry the principle of identity supplied by a unique `kind` (for example, a `kind` Person can have the `subkinds` Man and Woman that carry the principle of identity provided by Person). In particular, the term `Kind` is used to refer to `ultimate rigid sortals` (that is, the top-most rigid sortal in a given taxonomic tree) whose instances are objects (that is, functional complexes). Within the anti-rigid types, there is a further distinction between `phases` and `roles`. Both are specializations of rigid sortals. However, they are differentiated with respect to their specialization conditions. `Phases` are relationally independent types defined as a partition of a sortal. This partition is derived based on an intrinsic property of that type (for example, Child is a `phase` of Person, instantiated by instances of Person who are less than 12 years old). `Roles` are relationally dependent universals, capturing relational properties shared by instances of a given type. In other words, entities play roles when related to other entities For instance, a Person plays the `role` of Student when enrolled in a given Educational Institution (GUIZZARDI, 2005a).

UFO makes a fundamental distinction between two types of relations, namely: `formal` and `material` relations. Whilst the former holds directly between two entities without any further intervening individual, the latter is induced by the presence of mediating entities called `relators`. `Relators` are individuals with the power of connecting entities. For example, an Enrollment connects a Student with an Educational Institution. Every instance of a `relator` is existentially dependent of at least two distinct entities. The `formal` relations that take place between a `relator` and the object classes it mediates are termed `mediation` relations (GUIZZARDI, 2005a).

`Role` is strongly related to the notion of relations, since it represents relationally dependent sortal types. For instance, the `role` Student is played by a Person, when s/he is enrolled in an Educational Institution (GUIZZARDI, 2005a).

A special type of relation is the whole-part relation. A metaproperty used to distinguish among the categories of whole-part relations is *existential dependence*. An entity x is existentially dependent on another entity y iff in every situation that x exists then y must exist. Associated with existential dependence there is the notion of *generic dependence*. An entity x is generically dependent on a type Y iff in every situation where x exists an instance of Y must exist. As examples, there is the relation between a Heart and a Person (parthood with generic dependence), on one hand, and between a Brain and a particular Person (parthood with existential dependence), on the other: while a Heart must be part of an instance of *Person* (who does not have to be the same in every possible situation), a Brain needs to be part of a specific Person in all situations in which it exists (GUIZZARDI, 2005a).

Another metaproperty associated to whole-part relations is *shareability*. A (whole) type x is characterized by an exclusive (non-shareable) parthood relation with a (part) of type Y iff every instance of x must have at most one instance of Y as part. On the other hand, a type x is characterized by a shareable parthood relation with a type Y iff instances of x can have more than one instance of Y as part (GUIZZARDI, 2005a).

As said before, OntoUML is a DIML that consists of a set of stereotypes representing the types of the `universal` category of UFO-A. This language is an extension of the UML 2.0 class diagram proposed in (GUIZZARDI, 2005a). The language was elaborated through a process in which the UML metamodel was updated to guarantee an isomorphism of its structure to the reference ontology (UFO-A) and the axiomatization of UFO-A was transferred to the metamodel through formal constraints added to the metamodel. So, OntoUML, is an ontologically well-founded UML profile for class diagrams.

Regarding the taxonomy involving `individuals`, we highlight the concept of `qua-individuals - moments` that arise inherent to their bearers when they instantiate certain concepts or relations. In particular, the `qua-individuals` linked to `relators`, which are mediators `individual` (`individuals` who mediate others making a relationship true between them). An `individual relator` is composed of the relational `qua-individuals` that are inherent to the mediated `individuals` (the `roles`) when instantiating the relationship it represents (GUIZZARDI, 2005b). Also, it is inherent to the meriological sum of these `qua-individuals`. An example of an instantiation involving a relation (the Enrollment of a Student in a School Destination) and the relational `qua-individuals` is depicted in Figure 13.

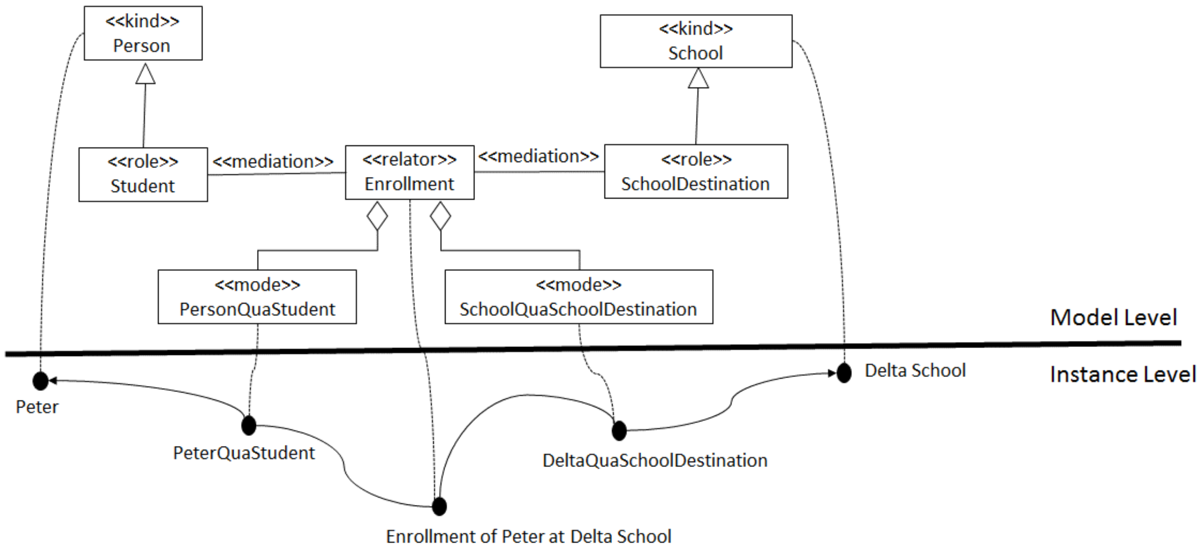


Figure 13: An example of `qua-individuals`

2.3.2 Ontological Guidelines for Designing VMLs

Guizzardi (2013) has established a connection between some OntoUML constructs and visual variables, taking PoN principles into account. He highlights two quality criteria in the design and evaluation of VMLs: (i) Domain appropriateness, which refers to truthfulness of the language to the domain; (ii) Comprehensibility appropriateness, which refers to the pragmatic efficiency of the language to support communication, understanding and reasoning in the problem domain. Domain appropriateness is more connected to the mapping between the real-world and the language abstract syntax. Comprehensibility appropriateness, in turn, has a strong connection to the language concrete syntax. Both quality criteria are expressed through the support of ontological theories as a way to improve quality in VMLs.

A direct influence that an ontology has over visual notations is due to the Semiotic Clarity principle, as Figure 14 shows.

Guizzardi (2013) argues that this principle is strongly connected to ontological analysis, a claim already done by Moody (MOODY, 2009). In its left side, the figure presents the relation between the language metamodel and the reference ontology. In the right side, the figure shows the relation between the metamodel and the concrete syntax. If the concepts of the reference ontology are used to evaluate the modeling elements of a language, which in turn are used to evaluate the representational elements, it can be said that the quality evaluation of the concrete syntax with respect to Semiotic Clarity indirectly but essentially depends on the characteristics of the underlying reference ontology.

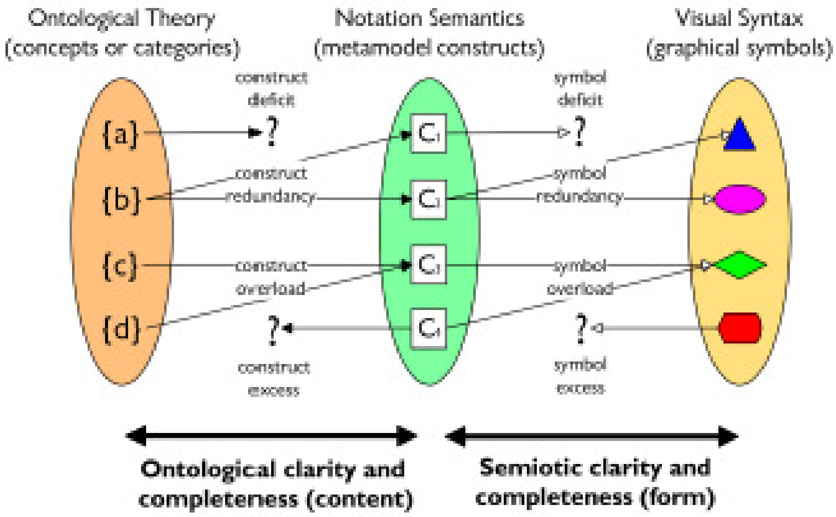


Figure 14: Construct anomalies and symbol anomalies relations in modeling languages. Source: (GUIZZARDI, 2013, p. 19), adapted from (MOODY, 2009, p. 759)

Another claim of Guizzardi is that if an isomorphism can be established between the structure of the modeling elements and the corresponding structure of the representational elements, it increases the transparency of the semantics of the latter. In other words, the visual notation elements should take the ontological categories and metaproperties of the reference ontology into account. Figure 15 presents an example, in which an isomorphism between elements in the domain and the visual representation of these elements is reinforced.

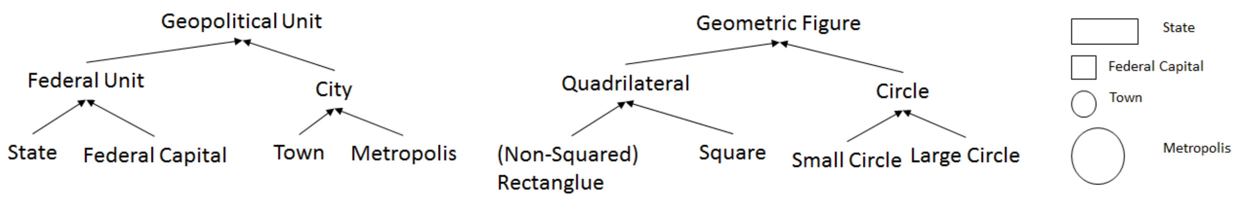


Figure 15: (a-left) a fragment of a taxonomy for a geopolitical domain; (b-center) a taxonomy of geometric objects isomorphic to the structure in (a); (c) a system of visual symbols from (b) to represent the domain concepts in (a). Source: (GUIZZARDI, 2013, p. 20)

In addition, Guizzardi (2013) describes a mapping between some OntoUML constructs and visual variables, taking PoN principles into account. This mapping is described next.

Kinds and ***Subkinds*** – Because of the existence of `subkinds`, a `kind` can be an abstract entity (not directly instantiated). Thus, it is not mandatory that every `kind` element has a direct symbol to represent it. A `kind` can be indirectly represented through the representational elements that represent its `subkinds`.

Shapes defined by closed contour are among the most basic metaphorical representations for object types. Moreover, a human most primitive notion of object is the notion of a maximally-topologically-self-connected object that moves in a spatial temporal trajectory together with all its parts. This idea is directly represented by convex shapes with closed boundaries (WARE, 2008). So, Guizzardi recommends mapping each concrete `(sub)kind` to a convex shape in this case. Moreover, the representational elements should be similar/dissimilar between each other in accordance to their corresponding meaning and taxonomical structure identified in the reference ontology, attending to the Perceptual Discriminability principle.

Another recommendation is the establishment of metaphorical resemblance when mapping the representational elements to their corresponding modeling elements, such as to represent a Man as a stick man. Sometimes the mapping is direct (through an icon), as in the Man example, and sometimes it is subtler (through a mnemonic or abstract symbol). Even the boundaries of a symbol can be used to reinforce a concept, as, for example, applying the visual variable texture to differentiate between more rigid / flexible concepts. These characteristics are in accordance to the Semantic Transparency principle.

***Phases* –** `Phases` are contingent specializations of `kinds` in such a manner that the specialization condition is due to changes of values in the intrinsic properties of the instances. So, the representation of this type should respect this aspect. It should be a variation respecting the shape that is used to represent the core concept (`(sub)Kind`). In this manner, the visual identity is kept, but the model reader can identify some type of change on it, characterizing the variation of `phases`.

Another recommendation is the application of color (a visual variable) values. Other visual variables, as brightness and texture, are also good options. A common decision is the use of high-saturation color aspect (brightness) to represent extreme situations exploring a metaphorical relation between “more quantity of color” and “more quantity of something” (for example, in an overloading situation). This is in conformance to Semantic Transparency and Perceptual Discriminability principles.

Even more, these visual variables can be applied in a complementary manner, as in situations in which it is interesting to highlight some concept, not only in its shape but in the shape boundaries – a clear application of redundant coding (a Perceptual Discriminability criterion).

***Relations* –** To represent whole-part relations, the use of spatial inclusion (position visual variable) is recommended. Each involved element should previously identify a symbol to represent it and these symbols are preserved in the representation of the relation.

`ComponentOf` relations are irreflexive and symmetric. Thus, by using the relation of spatial inclusion in the plane to represent these relations, we have a mapping to a visual relation that has exactly the same formal properties of the represented one, since spatial inclusion is also a partial order relation (GUIZZARDI, 2013). Also, this refers to a direct inferential free ride, that is, if A is `componentOf` B and B is `componentOf` C, then A is `componentOf` C (transitivity property).

To represent the non-shareability metaproperty, a good design decision is to apply semantic immediacy (a Semantic Transparency criterion) using signs comprised of tessellations (with non-overlapping regions). In this way, a model reader can easily notice to which whole element a part element is part of, besides directly visualizing the relation between whole and part elements. Even more, in a situation in which more than one `componentOf` relations are involved, the reader could visualize the manifestation of the transitivity metaproperty for this parthood relation. Also, this decision allows a clear comprehensibility of the `existential dependence` relation from the part to the whole - in favor of semantic immediacy. On the other hand, to represent the shareability property of some whole-part relation, it is suggested to apply spatial inclusions that allow overlapping regions, referring to the fact that a part element can be part of more than one whole element.

A common relationship between elements in a model is the superior-subordinate relation (or a similar one). In this case, it is suggested to represent this relation as, for example, an above relation in the plane (position visual variable), reinforcing semantic immediacy aspect. This is complemented by a common representation of relations between elements: a line connecting the involved elements. If a designer decides to represent more than one relation through line connection, s/he can apply different texture values to differentiate between the relations (lines connection), in accordance to Perceptual Discriminability principle.

Roles and Relational properties – `Roles` are specializations of `(sub)kinds` that have a relation specialization condition. If r1 is a `role` played by elements of `kind` k1 associated via the relation re to a `role` r2 played by elements of `kind` k2, then r1 and r2 should be represented by a representation relation between k1 and re, and k2 and r2, respectively. This implies that `roles` could be represented indirectly through the representation of the relation between the `kinds` involved in it, hence, in accordance to the Graphic Economy principle.

2.4 Final Considerations

Most works regarding CML design focus on the semantic aspect through the abstract syntax. Few studies have focused on the concrete syntax. However, modeling languages have different components and they should be connected. Defining a good quality modeling language implies that all the language elements should possess quality characteristics, even if they have different weights. We argue that the most appropriate is to have a complete approach that addresses all the language aspects (as abstract syntax, concrete syntax, representation strategies), with something interconnecting these aspects. In our case, we claim that this binding can be achieved through ontological theories that can be bound to information visualization theories. We selected PoN and UFO to support our claim.

Even being a widespread approach in the design of visual notations, PoN still presents limitations. For instance, it does not prescribe a design process for applying its principles. This and other challenges for applying PoN in practice have been identified in a number of works that have voiced them in a number of alternative ways: (i) The application of PoN is not without effort (GENON *et al.*, 2010) (VAN DER LINDEN *et al.*, 2016); (ii) PoN needs an expansion to be operationalized (STORRLE; FISH, 2013); (iii) One of the most challenging tasks is to elicit the semantic constructs of the language itself. This task is complex and is not as detailed as it should be in PoN (GENON *et al.*, 2010); (iv) Considering that the amount of constructs in a modeling language can be significant, it is difficult to a designer to keep in mind the visualization of each construct and the whole set (STORRLE; FISH, 2013).

Based on these considerations, we conclude that PoN requires a method defining how its principles can be applied. In an analysis process the PoN principles can be applied in a more flexible order - the sequence for application of principles is not a central issue. However, in a design process, the sequence in which we apply the PoN principles has a large influence in the work. In Chapter 4, we present PoN-S, a method systematizing the application of PoN principles.

Also, we have identified other issues that should be taken into consideration for establishing a solid design process for VMLs. Modeling is about human cognitive process. So, it is important to keep in mind pragmatic issues. Some PoN principles are more connected than others, as Semantic Transparency and Perceptual Discriminability. This indicates that they should be dealt with together. The same applies to Complexity Management and Cognitive Integration principles. These aspects are also considered in PoN-S.

Finally, PoN does not clearly prescribe how to establish symbols that are connected to the real-world entities they are supposed to represent, including the structure of the domain concepts. Guizzardi (2013) demonstrates that a bond can be established between language concrete syntax and ontologies, connecting PoN principles and UFO ontological metaproperties. This connection is a first step to solve the aforementioned limitation. However, his work still presents limitations: (i) The guidelines are in the format “for this entity type or relation apply this design decision”. So, they are isolated guidelines and not part of a design process; (ii) The set of considered OntoUML constructs and, thus, the set of ontological distinctions considered is restricted. In Chapter 5, we try to overcome these gaps, improving the ontological guidelines proposed by Guizzardi (2013), and incorporating then to PoN-S, giving rise to PoNTO-S.

After the literature review presented in the current chapter and the indications elaborated from it (listed above), we decided to initiate our work throught exploratory empirical studies, to check these indications and probably formulate new ones. These empirical studies are described in Chapter 3.

Chapter 3. Exploratory Empirical Studies

This chapter presents some empirical studies performed to collect an in-depth comprehension of the main theories adopted in this dissertation, that are, the Physics of Notations (PoN) and the Unified Foundational Ontology (UFO) applied in the design of modeling languages concrete syntaxes. Such studies are exploratory experiments which should result in the specification of some requirements for both PON-S and PONTO-S (see Chapter 4 and Chapter 5), acting as a complement of the theoretical studies.

The studies were motivated by questions such as:

(i) Is there some advantage in using ontologies in modeling language engineering, focusing on language concrete syntax? This question is explored in Section 3.1.

(ii) Considering that modeling of relationships is a difficult task and usually there are various ways to represent a relation, is there an ideal manner to represent relations in a model? This issue is explored in Section 3.2.

(iii) What are the impressions of PoN users when applying such approach in modeling language engineering? This question is discussed in Section 3.3.

While the first two experiments applied model interpretation tasks, that are, tasks typically performed by the language user, the third study is focused on the task of language development, therefore a task performed by the language designer.

The first experiment compares the interpretation of models represented with a concrete syntax that takes into account ontologies (a domain specific notation) and another concrete syntax not based on ontologies (a domain independent notation). The second experiment compares different design decisions related to representation of relationships. The third experiment aims in designing a concrete syntax whose abstract syntax is based on ontologies. So, all the different languages involved somehow in the experiments are based on ontologies, even if in the third experiment this fact was not explored.

The performed studies were exploratory studies applied in the beginning of the research process and they were previous of a clear definition of the direction we would adopt. On the contrary, besides they have confirmed some of our assumptions, they directed us in our decisions.

After presenting the three empirical studies, in Section 3.4 we describe some final considerations resulting from the experiments. These evidences together with the main characteristics and limitations identified in Chapter 2 were used to guide our next decision, that is, to build the PoN-S and PoNTO-S approaches. Also, these evidences were used as basis in the development process to elaborate PoN-S and PoNTO-S.

3.1 Empirical Study 1: Can Ontologies Systematically Help in the Design of Domain-Specific Visual Languages?

This empirical study was published in (SILVA TEIXEIRA *et al.*, 2013). The DSML applied in the study (see Figure 16 and Table 2) was initially presented in (GUIZZARDI, 2013). We decided to apply a previously defined DSML instead of creating a new one to concentrate our efforts in observing the language application. Also, using the ontology described in (GUIZZARDI, 2013), developed applying PoN and ontological guidelines, we have a greater confidence in the quality of the DSML.

3.1.1 Objectives

The *goal* of the experiment is to collect indications about the use of the concrete syntax of a DSML. The *research hypothesis* is that the performance of participants in interpreting instance elements diagram using the domain-specific notation is better (according to response time and correctness of answers) than that made by participants interpreting instance elements diagram written in a generic notation.

The language is related to a specific domain and it is represented by two dialects: (i) Generic notation, based on the UML notation for diagrams of objects, therefore composed of abstract symbols; (ii) Specific notation, based on ontological and information visualization guidelines, consisting of icons, indexs and abstract symbols. The experiment compares the results of interpretation tasks using these two dialects.

3.1.2 Design

In this section, we describe the design of the empirical study performed. It is organized in two parts: (i) The description of the DSML; (ii) The design of the study.

3.1.2.1 Description of the DSML

Figure 16 presents a fragment of an ontology of Organizational Structures. In this ontology, Employee is a `role` played by a Person when it is member of a Department. A Person (an abstract type) is either Man or Woman. An Employee is part of exactly one Department (represented by the non-shareable association end). Since this is a generic dependence relation, Employees can move to different Departments. An Employee involved in a reports to relation can be subordinated to at least one other Employee who is its superior, or s/he can be the superior to at least one other Employee who is its subordinate. Then, the types Subordinate Employee and Superior Employee are `roles` played by Employees. Since Subordinate Employee is a `role`, an instance of this type can cease to be one, and for it to instantiate this `role`, there must exist another Employee instantiating the Superior Employee `role`. The same instance of Employee can simultaneously instantiate both `roles`, but not in the same reports to instance.

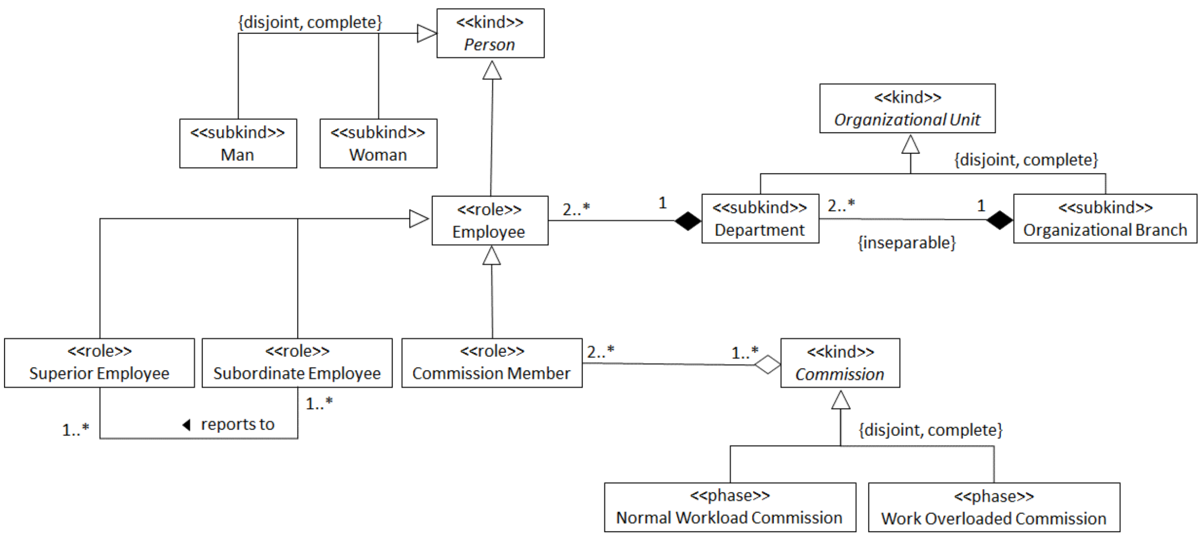
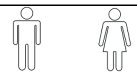
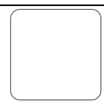
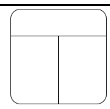
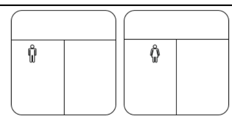
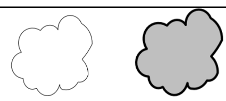
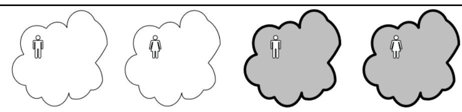


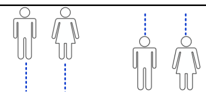
Figure 16: A fragment of an ontology for organizational structures. Source: (GUIZZARDI, 2013, p. 16)

A Department is part of exactly one Organizational Branch. Again, we have a case of a non-shareable parthood relation, but also one which implies existential dependency from part to whole (represented in OntoUML by the {inseparable} tag value), that is, the Sales Department of an Organizational Branch can only exist as part of that branch. The relations between Employee and Department, and Department and Organizational Branch are cases of transitive parthood as identified in (GUIZZARDI, 2009). Commissions are collectives that have particular Employees as members (termed Commission Members). Commissions can be in two different `phases` depending on the value of one of its intrinsic property (its amount of committed work). A Work-Overloaded Commission is a Commission such that its amount of committed work surpasses a certain threshold. A Normal Workload Commission is the complement of Commission with respect to Work-Overloaded Commission.

Based on the ontology of Figure 16, we designed a DSML concrete syntax aimed at representing valid instances of this ontology. This concrete syntax is the complement of the abstract syntax whose reference model is the ontology presented above. Table 2 presents the modeling primitives of this language via their respective concrete syntax. The table also relates these primitives with the domain concept they represent and with the ontological category of these domain concepts.

Table 2: Visual concrete syntax for the organization structure ontology of Figure 16

Domain Type	**Ontological Category**	**Representational Element**
Person, Organizational Unit, Commission	`Kind`	Abstract class; No direct representation
Man, Woman	`Subkind`	
Organizational Branch	`Subkind`	
Department and Department *is component of* Organizational Branch	`Subkind` and `Whole-part` relation	
Employee and Employee *is component of* Department	`Role` and `Whole-part` relation	
Normal Load Commission, Overloaded Commission	`Phase`	
Commission Member and Commission Member is part of Commission	`Role` and `Whole-part` relation	

Superior and Subordinate Employees	Role	
Subordinate Employee *reports to* Superior Employee	Domain association	Combination of *is-dashed-line-connected* with the *above* relation in the plane

This concrete syntax presents ***Semiotic Clarity***, that is, there is an isomorphic mapping[10] between the concepts in the domain ontology and the modeling primitives in the language. Moreover, the mapping between domain elements and elements in the visual notation takes full account of ontological categories and metaproperties of the former. Next, we elaborate on the systematic use of each of these ontological guidelines to derive properties of the concrete syntax.

Kinds and ***Subkinds***: In Figure 16, we have both kinds and subkinds. As discussed in (WARE, 2008), shapes defined by closed contour are among the most basic metaphorical representations for objects. This idea is in line with a number of findings in Cognitive Science, including the one where shape plays a fundamental role in kind classification (TVERSKY; HEMENWAY, 1984). In the language defined in Table 2, each concrete subkind is associated with a shape. The chosen shapes are sufficiently dissimilar and are aligned with the taxonomic relations between domain types as presented in Figure 16. For instance, the "four-sized" figures used to represent Organizational Branches and Departments are similar, considering they are Organizational Units. On the other hand, they are dissimilar from the blobs used to represent Commissions. These features highlight the principle of ***Perceptual Discriminability*** pointed by Moody and Hillegersberg (2009).

Another aspect is the direct metaphorical resemblance between the graphical elements used and their referents. One case is the iconic representation for Man and Woman. The representation of Departments as "pieces" of an Organizational Branch is adherent to the idea of "organizational divisions" associated to Departments. In addition, while the straight lines used in the contour of Organizational Units seems a more formal and rigid structure, the round boundaries of blobs representing Commissions are more naturally associated with a flexible informal one. The

[10] We do not consider that an abstract type interferes in the achievement of the isomorphic mapping, as it is a concept never instantiated directly in this proposal.

systematic use of these metaphorical resemblances brings to this notation another important quality characteristic according to Moody, namely, ***perceptual immediacy*** (MOODY; HILLEGERSBERG, 2009).

Phases: We used an intrinsic property of visual percept to represent different `phases` of a `kind` (the entity can change its `phase` but maintain its identity). In the concrete syntax presented in Table 2, the changes in color of blobs used to represent Commissions represent different `phases`. We use a high-saturation color to represent the Work-Overloaded Commission exploring a metaphorical relation between "more quantity of color" and "more quantity of work". This feature increases its ***perceptual immediacy***. The difference in brightness of grey hue used to represent an overloaded commission and white one used to represent a regular load commission creates an efficient ***perceptual pop-out*** (MOODY; HILLEGERSBERG, 2009). Finally, given that identifying overloaded commissions is an important task in the domain, the perceptual popout is increased by the increased ***perceptual discriminability*** between these two `phases`. This is due to the use of a different thickness of blobs boundaries. This is a case of ***redundant coding*** as pointed out by (MOODY; HILLEGERSBERG, 2009).

Relations: In the ontology of Figure 16, there are parthood relations between: (i) Employee and Department, and (ii) Department and Organizational Branch. They are irreflexive and asymmetric. Moreover, transitivity holds across (i) and (ii). By using the relation of spatial inclusion in the plane to represent these relations, we have a mapping to a visual relation that has exactly the same formal properties of the represented one, since spatial inclusion is also a partial order relation.

The different Departments that comprise an Organizational Branch are represented by a tessellation of the spatial region used to represent that branch. The lack of overlap between these regions allows for a ***perceptually immediate*** representation of the non-shareability metaproperty of these relations. This representation also contributes to ***perceptual immediacy*** due to the fact that, if Departments are represented as partitions of the region representing its associated Organizational Branches, this also favors the interpretation of existential dependence from part to whole.

Another parthood relation is between Commission Member and Commission. This relation is represented as a spatial containment relation between icons representing Person and a blob representing Commission. These blob forms can overlap with Department regions. This feature allows for direct inferential free ride on the identification of which Department a Commission Member belongs to. In addition, in line with the shareability metaproperty of this relation in the ontology, one can easily imagine overlapping blobs allowing for a certain member to be part of multiple commissions.

A third relation is the reports to relation, defined between a Superior Employee and its Subordinates. We used a combination of visual relations to represent this association (we combined the above relation in the plane with the transitive closure of the is-dashed-line-connected relation). Additionally, the different texture of this line increases the ***perceptual discriminability*** when contrasting it to the solid lines used to demarcate Department partitions. Finally, the spatial metaphor of using “higher in the plane” to represent “higher in the hierarchy” favors ***perceptual immediacy***.

Roles and ***Relational Properties***: Finally, we need visual representations able to highlight roles and their relational properties. The roles Employee and Commission Member are represented by the contained in the region relation between a Person icon and a region representing a Department and a Commission, respectively the roles Supervisor and Supervised *by* are represented by a dotted line between two Person icons and their spatial positioning.

3.1.2.2 Design

The empirical study was conducted following the guidelines presented in (JURISTO; MORENO, 2001).

The experiment has qualitative and quantitative *strategies*. The experimentation *level* is in-vitro (it was conducted in a controlled environment). The *research approach* is exploratory, to collect early indications for further studies. The experiment has as its *object of study* two instantiations of the conceptual model presented in Figure 16. The instantiations were presented in two different notations, a domain-specific notation and a generic notation, giving rise to four instance elements diagram.

The *subjects* are Computer Science students, from both under-graduate and post-graduate levels, which attend classes of a Conceptual Modeling course. The minimum requirement expected for participating in the experiment was having basic knowledge of UML. A questionnaire was applied to capture the participants profile. Regarding the sample size, there were 22 participants. They were divided into two groups (A and B) randomly. Group A had 12 participants, and Group B had 10 participants[11].

The *factor* of the experiment is the concrete syntax of a visual language, and the *alternatives* are: a generic notation (UML-based notation for object diagrams) and a domain-specific notation (presented in Table 2). The *task* is the interpretation of instance elements diagram for the same instantiations, using different notations. Questions regarding two instantiations of the conceptual model presented in Figure 16 were posed, varying the concrete syntax of the language used for representing them. The first instantiation represented using the domain-specific notation is depicted in Figure 17. A semantically equivalent representation using the generic notation is shown in Figure 18. Each participant had to answer two questions about this instantiation: one subjective (Q1), and another objective (Q2)[12]. Another instantiation, similar to the first one but larger, was also used and other two questions (Q3 and Q4) were posed. Q3 is subjective, while Q4 is objective. The questions and the predefined answers (separated by fragments) about the first instantiation are presented next. Regarding the second instantiation, Q3 has 6 answer fragments, and Q4 has 4 answer fragments.

1. *Consider the individual Lisa. What information can be obtained about this individual from the observation of the diagram?* Template Answer: Lisa is part of the Marketing Department (fragment 1). She is a woman (fragment 2). She is

---

[11] The imbalance in number of participants occurred, because, until the group drawing, we had 23 participants, and Group B (which had 11 participants) had one participant less for the effective execution of the activity.

[12] An objective question (as for example “how many” / “which is/are”) has a direct answer identifying one or more elements on the instantiation or it has a numerical answer. On the other hand, a subjective question demands an analysis of the connections of the elements identified in the question and make assumptions on them.

supervised by Mary (fragment 3). She supervises Ana (fragment 4) and Otto (fragment 5). She is member of the Quality Control Commission (fragment 6).

2. *Which is(are) the employee(s) with the largest number of direct subordinates*? *How many are the subordinates of this(these) employee(s)*? Template Answer: Peter (fragment 1) and Robert (fragment 2). They have three direct subordinates each (fragment 3).

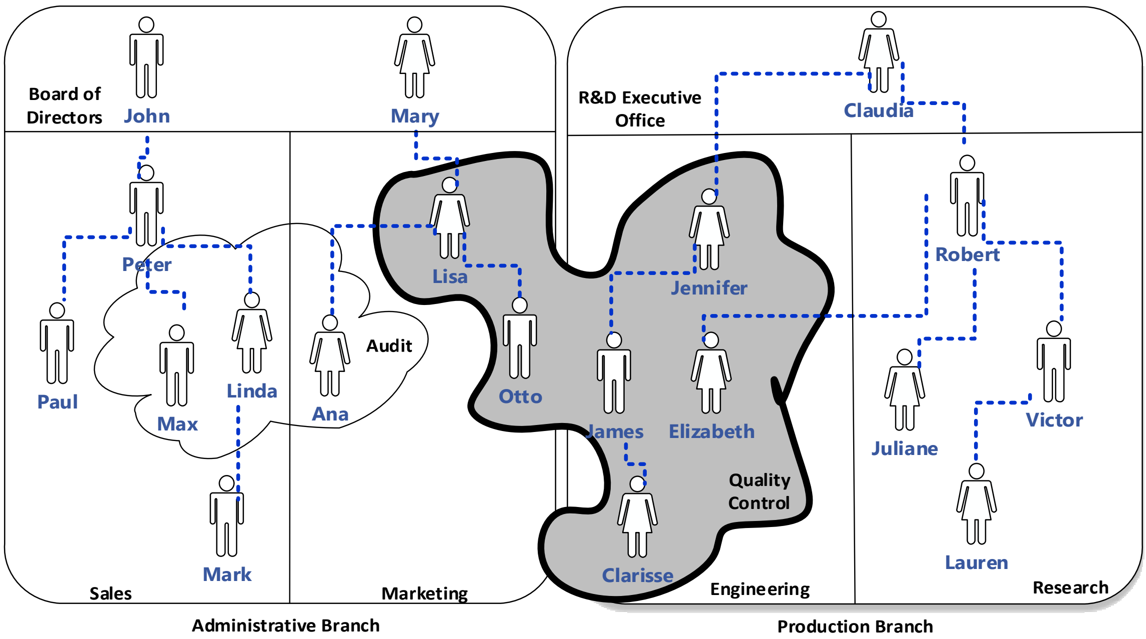


Figure 17: An instance elements diagram in the domain-specific notation

The dependent variables are: response time and correctness of the answers. These variables are measured for each question. The way to analyze response time is trivial: the time taken to answer each question is recorded and the smaller the time, the best is the quality of the notation used in the interpretation task. Correctness is measured by comparing the fragments of the participants answers to the corresponding fragments of the template. If they are the same, then the fragment is *correct*; otherwise the fragment is *wrong*, if the participant said something wrong about the fragment, or *missing*, if the fragment in the template is not reported by the participant. This is what we call *fragment correctness*. We have also other two types of fragments: wrong complementary fragment, which occurs when the participant included in her answer a fragment that the model does not say, related or not to the question; and extra complementary fragment, when the participant included in her answer a fragment that is not part of the template, but it can be inferred from the diagram. An extra complementary fragment is not a mistake, and therefore it is simply ignored.

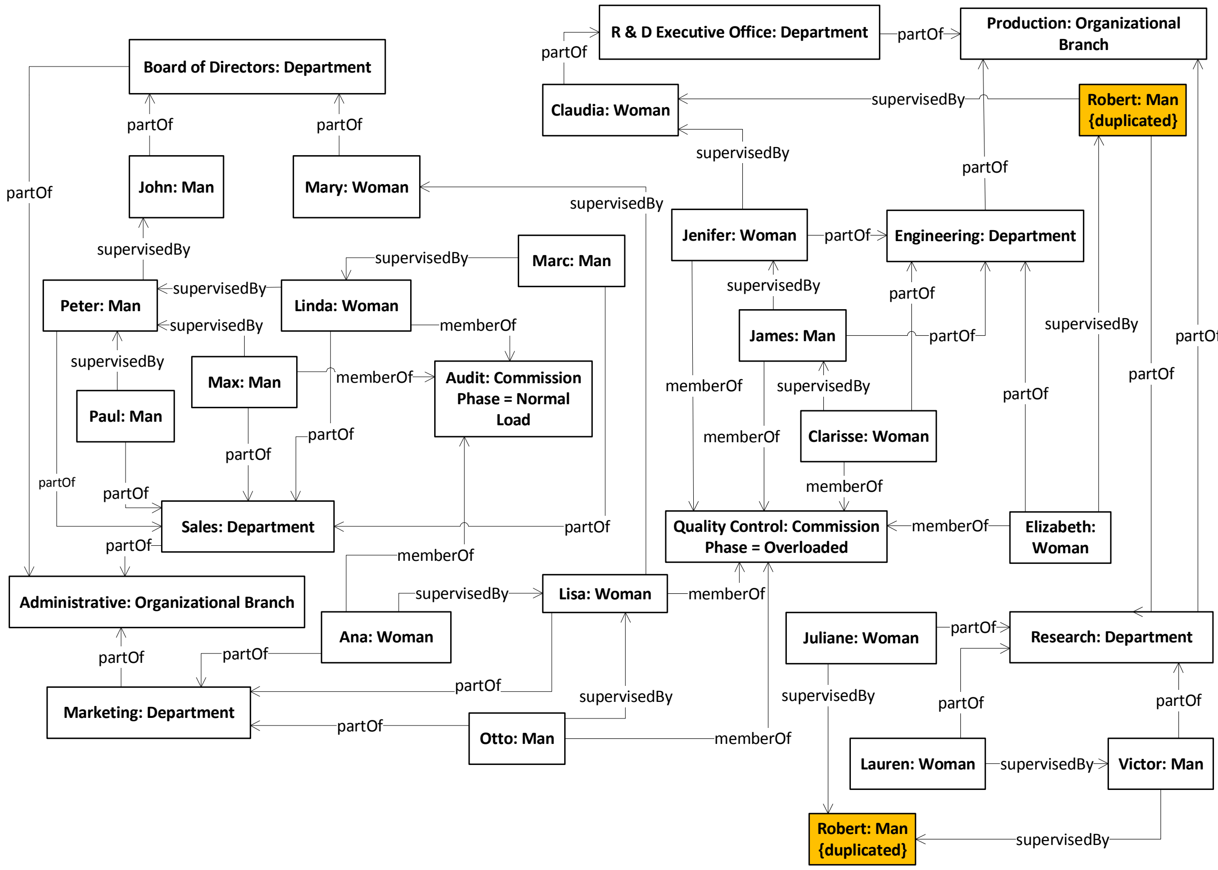


Figure 18: An instance elements diagram in the generic notation[13]

To illustrate how we analyzed fragment correctness, consider the following answer given by a participant to question Q1: "*Lisa is part of the* Board of Directors Department, and she *is supervised by Mary. She is also part of* the *Marketing Department and she supervises Ana and Otto*. *She is part of the Quality Control Commission"*. According to the template, there are five correct fragments (1, 3, 4, 5 and 6), one missing (fragment 2), and one wrong complement (*Lisa is part of the* Board of Directors *Department*).

Besides interpreting the diagrams, the participants filled in a questionnaire containing the following questions: (1) What is your impression (which one was easier / more difficult, better/worse) among the concrete syntaxes presented? (2) Add any additional observation that you deem necessary.

[13] The object Robert: Man is duplicated in the diagram to avoid crossing lines.

Each group answered the same questions by interpreting each instantiation in a different notation. Group A answered the two questions (1 and 2) of the first instantiation in the domain-specific notation, while Group B answered the same questions of this instantiation using the generic notation. In the second instantiation, the situation was reversed, for answering questions 3 and 4, Group A interpreted the diagram written in the generic notation, while Group B interpreted the diagram written in the domain-specific notation.

In order to facilitate data collection, a website was developed. The site contained the instance elements diagram, the corresponding questions, and a link to the notation used in each instantiation. We recorded the participants answers and the response time for each question automatically. Although we used a website, the experiment was conducted during a class in a lab, in order to ensure a stable Internet connection and to avoid distractions to participants, thereby reducing threats to the experiment.

3.1.3 Results

Regarding the participants profile, we state that: (i) The educational level (undergraduate, master and doctoral students) of the groups were balanced; (ii) Regarding experience time in conceptual modeling, Group A had around 90% of its members with experience above 1 year, while Group B had 80% of participants in this range. We consider that the groups were balanced, even if members of Group A had a little more experience. The participants profile was of students acting as modelers with some knowledge in conceptual modeling, specifically on using UML.

Table 3 presents data regarding the response time for each question. The columns present data on average, median, highest and lowest value of response time, and the percentage difference between highest and lowest averages for a question. Table 4 shows the percentage of fragment correctness for each question, and also the number of wrong complementary fragments, grouped by notation. In Table 3 and Table 4 we highlighted in grey the items that are consistent with our hypothesis, and in black the ones that contradict our hypothesis. Figure 19 presents four graphs showing the response times for each question, comparing the values generated by the use of each notation in each question. The values are ordered[14].

Table 3: Response time (in seconds)

Question	Average (av)		Median		Highest Value		Lowest Value		(Smallest av / Largest av)
	Gr. A	Gr. B	Gr. A	Gr. B	Gr. A	Gr. B	Gr. A	Gr. B	
Q1	363,25	301,67	346,5	292	715	463	148	141	83,05%
Q2	101,92	210,22	99	226	157	324	49	79	48,48%
Q3	400,67	271,67	319,5	260	1416	453	153	117	67,80%
Q4	210,50	62,67	206,5	61	512	99	83	35	29,77%

Table 4: Percentage of correct, wrong and missing fragments, and number of wrong complementary fragments by notation

Question	% of Correct Fragments		% of Wrong Fragments		% of Missing Fragments		Number of Wrong Complementary Fragments	
	Specific	Generic	Specific	Generic	Specific	Generic	Specific	Generic
Q1	66,67%	85,00%	0,00%	3,33%	33,33%	11,67%	4	2
Q2	86,11%	90,0%	11,11%	3,33%	2,78%	6,67%	0	0
Q3	71,67%	88,89%	0,00%	0,00%	28,33%	11,11%	11	3
Q4	97,50%	75,00%	2,50%	12,50%	0,00%	12,50%	0	5

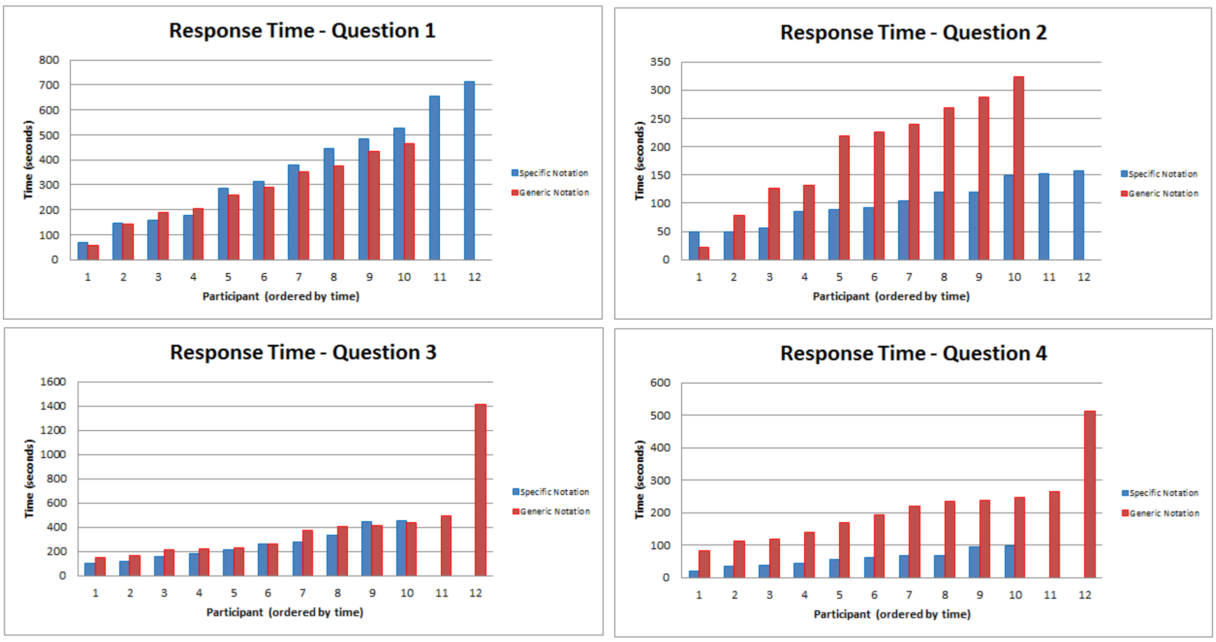


Figure 19: Evolution of response time per question for each notation

[14] Each group of response time is ordereded from the smallest to the highest time. In this way, the graphics are comparing the smallest time of the generic notation with the smallest time of the specific notation in each question, and so on.

Figure 20 shows eight graphs showing the number of correct, wrong, missing, and wrong complementary fragments, two per question, comparing the notations. The values are ordered by number of correct fragments.

Figure 20: Number of correct, wrong, missing and wrong complementary fragments per question, for each notation

3.1.4 Discussion

According to our hypothesis (see Section 3.1.1), Group A should perform better on Q1 and Q2, while Group B should better perform on Q3 and Q4, as these are the moments that each group works with the domain-specific notation. However, this was not always the case.

Looking at Table 3 and Figure 19, we can say that, regarding response time, the expected results are confirmed for Q2, Q3 and Q4; however, the results contradicted our hypothesis for Q1. Moreover, the percentage differences between highest and lowest averages were greatly varied. For Q1, highest and lowest averages had nearest values, while for Q4 they presented the highest difference. The intention of generating such values is to observe how, on average, response times are different according to the notation. It is expected that the differences are significant, as occurred in Q2 and Q4, and even in Q3, favoring our hypothesis.

We have applied a statistical test, even having worked with a small sample. We applied the Wilcoxon-Mann-Whitney U Test (WADE; KOUTOUMANOU, 2010), with significance level of 5%, for comparing response times. Considering groups A and B, U test indicated that the values are not significantly different, which is a good first indication that the groups are balanced. In Q1 and Q2, U test indicated that the values are not significantly different among groups, which was probably caused by the result of Q1. For Q3 and Q4, U test indicated that the values are significantly different among groups, what is a favorable result to our hypothesis.

However, an insulated analysis of response time is not enough. We need to check whether these results are corroborated by fragment correctness.

Looking at Table 4, which summarizes fragment correctness, and Figure 20, we can notice that, again, we achieved results that are in favor and that contradict our hypothesis. Observing the number of correct fragments, we realize that specific notation worked better in Q4, but generic notation worked better in the other questions, with a small difference in Q2 (less than 1%). The most significant percentage of correctness occurred in Q4, and the lowest in Q1 (both from specific notation). Overall, the average correctness percentage is high (above 66%), demonstrating that participants had mostly success in the interpretation of models. On the number of wrong fragments, we had a slightly different result. In Q1 and Q4 we had fewer wrong fragments when using the specific notation, while in Q3 we had fewer wrong fragments when using the generic notation, in Q2 both had none error. Regarding missing fragments, the specific notation worked better for the objective questions (Q2 and Q4), while the generic notation worked better for the subjective questions (Q1 and Q3). Moreover, in the generic notation there is a relatively stable percentage (between 6% and 13%) of missing fragments. In the case of the specific notation, however, there is a stark difference when comparing objective and subjective questions. For objective questions, the percentages of missing fragments are very low (less than 3% for Q2, and 0% for Q4). On the other hand, for subjective questions, the percentages of missing fragments are very high (about 30% for Q1, and about 25% for Q3).

Finally, regarding the number of wrong complementary fragments, we can notice that the specific notation worked better for the objective questions (Q4), while the generic notation worked better for the subjective questions (Q1 and Q3). We should also highlight that, in Q2, there is not any wrong complementary fragment in both notations. Moreover, in the answers for Q3, when interpreting the model written in the specific notation, there is a high number of wrong complementary fragments.

Next, we complement our analysis presenting some detailed information for each question.

Question 1. The results obtained in this question clearly contradict our hypothesis. Q1 is a subjective question, requiring inspecting in detail a model element. It requires a great attention by the participants, since it contains the greatest number of different fragments in the expected response. Response times were close, with a slight advantage for the generic notation. The missing parts stood out significantly in the domain specific notation. For instance, 10 participants did not indicate in the domain-specific notation that Lisa is a woman (83%), while only 6 participants in the generic notation did not indicate this fact (60%). Maybe it was considered obvious in the former notation, while in the case of the generic notation, perhaps the difficulty was lower, since this information is written in the model (allowing a textual perception). It is interesting to notice that, in fact, it was the number of missing fragments that caused the hypothesis contradiction, since the number of wrong fragments is favorable to the domain-specific notation. There were situations where the information was given only partially (counting as missing). For instance, there were an answer indicating that an employee supervises someone, but without indicating who is the supervised employee. Regarding the number of wrong complementary fragments, based on the answers, we suppose that some participants were not actually aware of the notation (commission treated as a kind of department or as a group).

Question 2. Q2 is an objective question. The domain-specific notation had a better performance regarding response time and the number of missing fragments. However, a better performance is achieved when using the generic notation with respect to the number of correct (a small difference) and wrong fragments. A possible explanation for this result came from the interpretation of a participant to this question: instead of answering who is the employee with the highest number of direct subordinates, she identified the employee with the greatest number of direct and indirect subordinates, giving rise to three wrong fragments. Once we worked with a small sample, this fact had a significant impact in the result. If we considered such participant an outlier, the result would have been reversed.

Question 3. Like Q1, this is a subjective question, and thus the results are similar: response times are close (with a small advantage for the domain-specific notation, as opposed to Q1), and there is a general advantage of the generic notation regarding correctness. Again, the number of missing fragments is able to change the outcome. In the specific notation, two participants indicated that there are 5 employees in the department, without indicating who are them (accounting for 10 missing fragments). In the generic notation, only one participant made this mistake. Moreover, in the specific notation, five participants did not indicate that the Marketing Department is part of the Administrative branch, while in the generic notation only one participant made this mistake. It is worthwhile to point out that the answers for this question presented the highest number of wrong complementary fragments in the experiment, highlighting the case of saying that Lisa is the leader of the Marketing Department (3 occurrences in generic notation and 5 occurrences in domain-specific notation).

Question 4. This was the question with the greatest proximity to our hypothesis. Specific notation presented better response time, higher percentage of correct fragments, and lower percentage of both wrong and missing fragments (in fact, just one error). This is a question that requires the participants to realize which are the members of a commission, and then to which departments they belong. In the specific notation, this is easy to notice, since a person is within both the regions representing the commission and the department. On the other hand, in the generic notation, it is harder to follow the lines connecting the elements. Moreover, it is interesting to notice that the graph for this question when using the domain-specific notation is quite similar to the one for Q2, especially if we ignore the outlier in the latter.

In summary, the most prominent indication we noticed is that the participants using the domain-specific notation performed better on objective questions, while participants using the generic notation performed better in subjective questions. However, not all signs can be confirmed or justified. Nevertheless, we can say that the experiment fulfilled the objective of generating evidence in a qualitative way, being the starting point for subsequent experiments.

Some indications obtained from the results that need to be further explored: (i) Do the familiarity with the notation based on UML (generic one) may have assisted in the analysis of diagrams, closing, or even exceeding, the performance of the domain-specific notation, with which participants had the first contact? (ii) In the generic notation, all information about a given object are obtained in the same way: by navigating through links between objects, while with a DSML, there are different ways of obtaining such information. How do this affect the results? (iii) Why in subjective questions is there a significant occurrence of missing fragments? Do not the participants perceive the information in the model, or at least do not feel the need to record the information, which can, for example, be considered "obvious"?

3.1.5 Limitations and Validity Threats

During the experiment, we identified some limitations and validity threats.

- The participants of the experiment were students with different profiles. So, the groups could have an imbalance of experience levels, which can impact in the interpretation task. However, we analyzed the profile of each group and we concluded that this threat has not been materialized, even if Group A was a bit more experienced than Group B;
- Background knowledge in UML may have caused an increase of correct answers when interpreting diagrams in the generic notation compared to the results achieved when applying the domain-specific notation, with which the participants had the first contact. We did not anticipate such a threat and did not take a precaution against it. It is an evidence to be evaluated in a later study;

- The number of participants was small, and thus we had not a representative sample of the target public (professional language users of the proposed DSML performing interpretation tasks of instance elements diagrams);
- We consider the interpretation of small fragments of conceptual models. The participant comprehensibility in larger models can lead to different results than we realized at this early stage. We intend to investigate this possibility;
- Complementing the item above, we worked with a DSML of common sense and relatively simple domain. The participant interpretation of more complex and larger domains could be different.

3.2 Empirical Study 2: Analyzing the Behavior of Modelers in Interpreting Relationships in Conceptual Models

This empirical study was published in (SILVA TEIXEIRA *et al.*, 2014). It was motivated by our need to collect evidence on the most appropriate way of representing relations, and corroborate researches that analyzed the difficulty of modeling relationships, including ontological analysis, and the several ways to represent them, as exposed in (WAND *et al.*, 1999) and (GUIZZARDI; WAGNER, 2008b).

3.2.1 Objectives

The experiment *goal* is to collect indications of the impact of different *representation strategies* for modeling relationships in the interpretation of fragments of type elements diagrams. The *research hypothesis* is that the presence of `relators` and `roles` increases the diagram clarity and, hence, improves the performance of model readers in interpreting the fragments using both constructs (considering correctness of answers according to a template, as well as the rationale given by the participants) when compared to the behavior of participants interpreting the fragments without them.

3.2.2 Design

The experiment was conducted following the guidelines presented in (JURISTO; MORENO, 2001).

The *objects of study* are conceptual model fragments developed using OntoUML in different domains. The experiment has qualitative and quantitative strategies. The *experimentation level* was in-vitro (it was conducted in a controlled environment). The *research approach* is primarily analytical, to collect early indications for further experiments. The first diagram fragment focuses on representing a reflexive relationship, while the second focuses on representing a binary relationship between two different concepts.

The *subjects* are Computer Science students, which attend classes of a Conceptual Modeling course. The minimum requirement expected for participating in the experiment is having basic knowledge of UML and OntoUML. A questionnaire was applied to capture the participants profile. Regarding the sample size, there were 22 participants. They were divided into four groups (GA, GB, GC and GD) randomly. GA and GB had 6 participants each, and GC and GD had 5 participants each.

The participants profile is of students acting as model users (readers) with some knowledge in conceptual modeling, specifically on using UML and OntoUML relations. The educational level (undergraduate, master and doctoral students) of the four groups was balanced (most participants were postgraduate students, but all groups have at least one undergraduate student). Regarding experience time in conceptual modeling using UML, all members of GA and GC have more than one year of experience, while GB has around 80% of its members in this range, and GD has around 60% of its members in this range. Concerning knowledge in OntoUML prior to the Conceptual Modeling course, around 80% of the participants in GA and GC already had prior knowledge in the language. That number drops to around 30% in GB and 40% in GD. However, over the course, OntoUML was studied and all the participants had access to the minimum knowledge necessary for performing this activity. At first, we considered that this would cause equilibrium between the distinct groups. The results, however, showed us that this was not the case, and the level of experience of the participants influenced their interpretation of the diagrams.

The *factor* of the experiment is the representation of relationships using `relator` and `role` constructs in diagrams. The alternatives are: (i) Representing only the `material` relation, without ***`relator`*** and ***`role`*** (see Figure 21, Group A); (ii) Representing `roles` and `material` relation, without `relator` (see Figure 21, Group B and Figure 22, Group C); (iii) Representing `relator`, `roles` and the corresponding `mediation` relations (see Figure 21, Group D and Figure 22, Group B); (iv) Representing `relator` and the corresponding `mediation` relations, without `roles` (see Figure 22, Group D); (v) representing `role`, `relator`, the corresponding `mediation` relations, and the `material` relation derived from the `relator` (henceforth termed "complete representation") (see Figure 21, Group C and Figure 22, Group A). The task is to interpret two diagram fragments, each one regarding a different domain, using different representations. Figure 21 and Figure 22 depict the representations of the diagram fragments for each domain and group. The interpretation is done by means of answering four questions relative to each fragment. Each response should include an explanation of how the participant arrived at the answer.

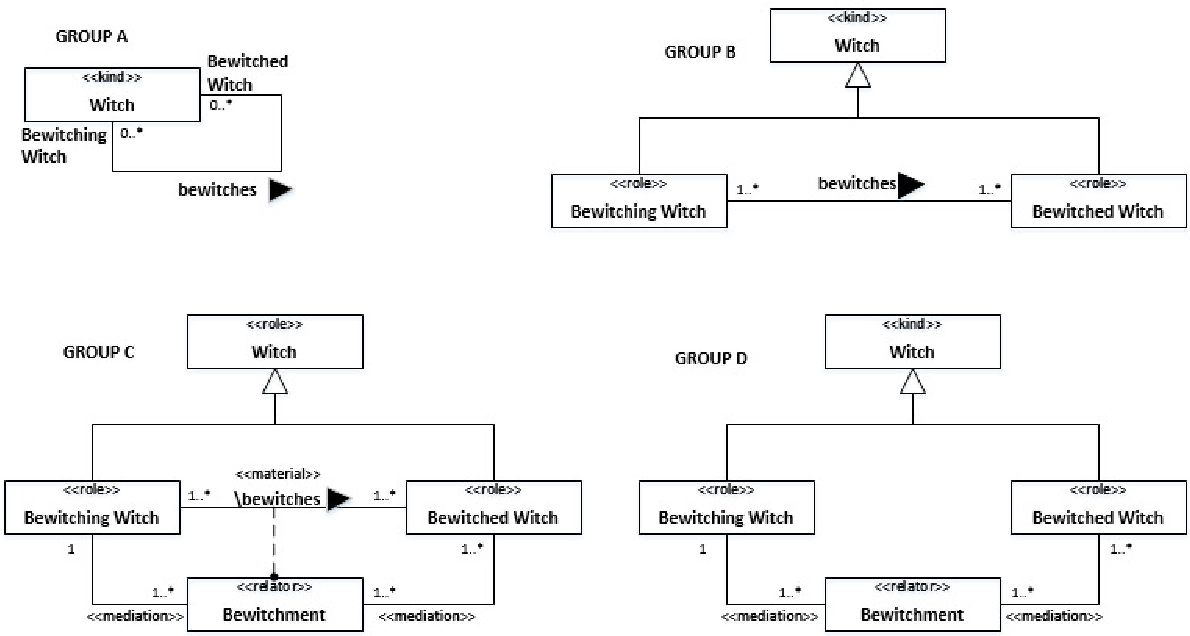


Figure 21: Diagram fragments of domain 1 interpreted by each group

The selected domains were artificially designed. In the first domain, the type elements diagram fragments intend to capture a relation between <u>Witches</u> involved in a <u>Bewitchment</u>. In this fictional world, witches can put spells (<u>Bewitchments</u>) on each other. In the second domain, the fragments are designed to represent a relation between a Knight and a Tamed Dragon. In this (once more) fictitious domain, Knights and their Tamed Dragons are connected by a Loyalty Bond. The intention behind the choice of type elements diagram fragments in fictional domains has the goal of minimizing the use of previous domain knowledge in answering the questions.

The questions, which are the same for all groups, are:

- Domain 1 (Bewitching Witches bewitch Bewitched Witches): Q1) How many Witches can be bewitched by a Bewitchment? Q2) How many Witches can be the Bewitching Witch of a Bewitchment? Q3) Can a Witch be bewitched by her own Bewitchment? Q4) Can a Bewitchment exist without affecting any Bewitched Witch?
- Domain 2 (Bonds between Knights and Tamed Dragons): Q1) Can a Tamed Dragon exist without a Knight? Q2) How many Bonds might exist between a Dragon and a Knight? Q3) Can a Person be a Knight without a Bond with a Dragon? Q4) How many Bonds a Knight may participate at the same time?

Q2 in Domain 2 includes an implicit condition being tested. Instead of asking for Tamed Dragon (a `role`) it asks for Dragon (a `kind`). There is nothing in the diagram specifying that all dragons must be tamed. We wanted to verify if the participants are aware of this difference.

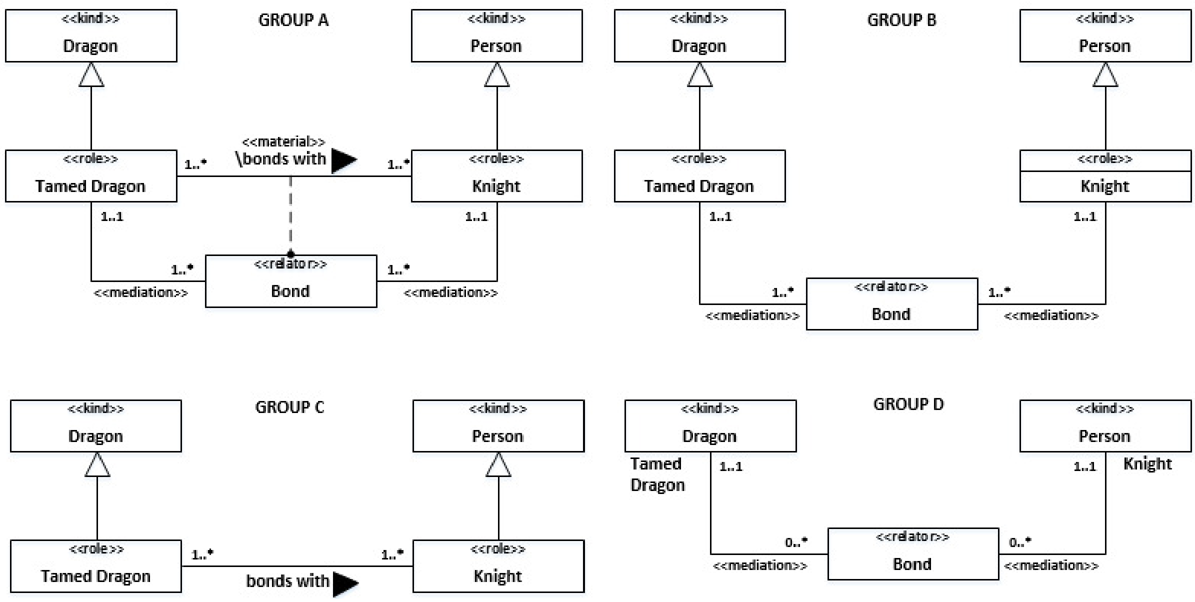


Figure 22: Diagram fragments of domain 2 interpreted by each group

The dependent variable is correctness of answers, considering also the rationale followed by the participant. This variable is measured by comparing the fragments answer of the participants answers to the corresponding fragments of a template. If they are the same, then the fragment answer is correct; otherwise, the fragment answer is considered incorrect.

3.2.3 Results

Table 5 presents the tabulated results in terms of percentage of correct answers for each question per group. As we can notice, GA and GC (groups containing the most experienced participants) presented better performance (over 50% of correct answers), than GB and GD (less experienced groups). These latter groups answered less than 50% of the questions right. In both domains, the highest success rate (100% correct) was obtained by groups GC and GA. We had also cases in which no correct answers were given (0% correct). The latter case happened in groups GA, GC and GD. It is worth to remember that Q2 of Domain 2 had an implicit condition being tested. This could justify the poor performance of the groups, including GA and GC.

Table 5: Correctness of answers (aproximated percentage - %)

Domain	Group															
	Question 1				Question 2				Question 3				Question 4			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
1	33	16	100	40	33	33	100	60	100	83	20	0	16	33	100	80
2	100	83	100	20	0	16	0	40	100	66	60	20	50	33	0	0
Tot.	GA: 54.17				GB: 45.83				GC: 60.0				GD: 32.5			
Rel*	GA: 54.17				GB: 29.17				GC: 26.67				GD: -			

Rel* = ***Relator*** Absence Identified (complementary information).

Another aspect useful for evaluating the groups´ performance are the cases of participants who indicated difficulties in answering a question due to the absence of a `relator` (see Table 5). For instance, consider Q1 in Domain 1. In our template, we established as an ideal response: Minimum = “1”. Maximum = “it cannot be determined with certainty”. An instance of the relationship bewitches must exist for the `roles` Bewitching Witch and Bewitched Witch to exist too. If such relationship exists, then there is at least one Bewitching Witch and one Bewitched Witch instances. The relationship cardinality also allows to infer that the same Bewitching Witch can bewitch several Witchs (Bewitched Witch), but this may occur through one Bewitchment or even several Bewitchments - which cannot be deduced from the diagram where such concept is not present. We notice that, in comparison with group GB, a significant percentage of GA members indicated that the absence of an explicitly represented `relator` prevented them in answering these questions. An unexpected outcome was noticed in the behavior of GC (more experienced group), which has less than 30% of participants indicating difficulties to answer a question due to `relator` absence. It is worthwhile to point out that this aspect did not apply to GD, since all diagram fragments read by them contained `relators` in both domains.

Table 6 also presents the results regarding correctness, but focusing on the different *representation strategies*. We noticed that the best performance was achieved by groups using the complete representation. The second best result is when both `relator` and `roles` are presented, but the derived `material` relation is not. However, we should reinforce that, in general, we perceived an influence of the experience level in the results. Thus, it is more adequate to compare the performances of a same group using different representations. This comparison provides an indication about the influence of the presence/absence of `relator` and/or `role` constructs. For instance, consider GC. In the complete representation, GC obtained 80% of correct answers. However, using a representation without `relator`, the same group answered only 40% of the answers correctly.

Table 6: Average percentage of correct answers by representation strategy

Domain	Complete		With relator and roles / Without material relation		With relator / Without roles		Without relator / With roles		Without relator and roles	
	Group	%	Group	%	Group	%	Group	%	Group	%
1	C	80.00	D	45.00	-	-	B	41.67	A	45.83
2	A	62.50	B	50.00	D	20.00	C	40.00	-	-
Total		**70.45**		**47.73**		**20.00**		**40.91**		**45.83**

Table 7 shows an interesting aspect regarding the use of optional associations versus the use of `roles` and mandatory associations. This observation involves questions of type "how many" (Q1 and Q2 in D1; Q2 and Q4 in D2) concerning the minimum value indicated in the answers. To simplify we compare only two groups per domain. In D1, we compare GA (without `role`, with optional relationship) and GC (with `role`, with mandatory relationship). In D2, we compare GD (without `role`, with optional relationship) and GB (with `role`, with mandatory relationship). For these questions, the minimum cardinality should be 1, except for the case of Q2 in D2, which, as previously mentioned, has an implicit condition being tested, and because of this, the minimum 0 is acceptable. In both domains we noticed better performances by the groups that worked with representations that explicitly show the `role` construct.

Table 7: Percentage of responses in questions of the type "How many" separated by minimum cardinality indicated per group

Domain	**Question**	**GA**		**GB**		**GC**		**GD**	
		Min. 0	**Min. 1**	**Min. 0**	**Min. 1**	**Min. 0**	**Min. 1**	**Min. 0**	**Min. 1**
D1	**Q1**	66.67%	0.00%	-	-	0.00%	100.00%	-	-
D1	**Q2**	66.67%	0.00%	-	-	0.00%	100.00%	-	-
D2	**Q2**	-	-	16.67%	33.33%	-	-	40.00%	0.00%
D2	**Q4**	-	-	0.00%	66.67%	-	-	20.00%	0.00%

In general, the results of the experiment point out that: 1) The experience level interferes in the participants performance in the interpretation tasks. Most experienced groups presented better performance; 2) The representation strategy affects the participants performance in the tasks. Participants reading diagram fragments that explicitly show the constructs `relator` and `role` presented better performance.

For answering the experiment research questions, we used the collected data presented above, as well as analyses of participants responses. To illustrate these analyses, next we present the analysis done for question Q4 of domain D1. Similar analyses were performed on the other questions of the experiment, but only the main findings are identified here.

D1. Q4: Can a bewitchment exist without affecting any Bewitched Witch?

Table 8 presents the performance of the participants in this question.

Table 8: Performance of groups in D1.Q4

Group	**Clearly Correct Answers**	**Declared impossibility to precisely answer Q4**
A	33.33%	16.67%
B	50%	50%
C	100%	None
D	80%	None

In GA, 3 out of 6 answered the question incorrectly ("yes", due to the zero minimum cardinality constraint). The correct answer is: "no", given that a Bewitchment is an existentially dependent entity and it depends on at least one individual playing the `role` of a Bewitched Witch as well as at least one (distinct) individual playing the `role` of Bewitching Witch. A 4th participant answered that it was impossible to answer the question precisely without the explicit representation of the Bewitchment. Despite sensible, we consider this answer as incorrect since: (i) In order to be a Bewitched Witch one has to participate in at least one relationship; (ii) Since this is a `material` relation, participating in this relationship means being mediated by a `relator` (a Bewitchment); (iii) Given that a `relator` is an existentially dependent entity, it must connect at least one instance of Bewitched Witch (and at least one distinct instance of Bewitching Witch). A 5th participant answered the question stating that if a Bewitchment is interpreted as a pair, then it should contain an instance of Bewitched Witch in the pair. Notice that although showing a sensible reasoning, this answer highlights a conceptual mistake, namely, that a Bewitchment is an instance of the relationship. In contrast, given the cardinalities of the diagram, it is clearly possible for the same Bewitchment instance involves a number of Bewitching Witches and Bewitched Witches. Anyway, we consider in this context the answer as a correct one. A 6th participant answered: "the role indicates that an individual is a Bewitching Witch when participating in a relationship". We envisage that the participant followed the same reasoning of the previous one.

In GB, 3 out of 6 participants stated that it was impossible to answer the question. However, as explained about for the 4th participant of GA, we consider this answer incorrect. Three of the participants answered the question correctly. Moreover, one of them made an explicit reference to the `relator`, while the other two based their answer solely on the cardinality constraint (minimum of 1).

In GC, all participants answered the question uniformly and correctly. A similar tendency can be observed for GD in which 4 out of 5 participants gave the same correct answer. A 5th participant of GD mentioned that the diagram did not contain enough information for the question to be answered (which is an incorrect answer).

In summary, the groups with an explicit representation of the `relator` performed better in answering the question – observe GC and GD in Table 8. Moreover, the group with the complete representation (GC) was the only group with unanimously correct and justified answers. GA had the worst performance, with the majority of the participants showing a shallow reasoning process by reasoning only on the directly represented cardinality of the represented relation and, hence, answering the question incorrectly (50% of incorrect answers for this reason). The participants of GA and GB that answer the question correctly did so by reasoning on the instance of the relation and on the `role` at hand. It is no surprise that this performance was then better in GB where the `roles` were explicit represented (50% correct answers) than in GA (33.33%). Finally, half of the participants in GB and one participant in GA answered the question incorrectly by being unsuccessful in reconstructing the relation between `roles`, `material` relations, the `relator`. Furthermore, the only participant in GA that clearly answered the question correctly also made a conceptual mistake in equating the instance of the relation with the `relator`. This is a common mistake (see discussion in (GUIZZARDI; WAGNER, 2008a)). However, we interpret this mistake as also being influenced by the non-explicit representation of the `relator` and its connection to the `material` relation at hand.

3.2.4 Discussion

Next, we present the discussion through the research questions (RQ1 and RQ2 below) we elaborated on.

RQ1) What are the effects of optional relationship in interpreting model fragments?

Regarding the use of the `role` construct to prevent the occurrence of optional associations, we conclude that it leads to a better performance of the participants. This evidence can be perceived by contrasting diagram fragments with optional association (without `role`) and with mandatory association (with `role`), as Table 7 shows.

Contrasting groups with equivalent experience, in Domain 1, we see that GC had 100% of correct answers (minimum 1) in both questions (Q1 and Q2), while GA indicated more than 65% of incorrect answers (minimum 0) in both questions. The remainder of the responses were also incorrect, but due to some other reasons that we refrain from discussing here. This is a clear indication of the influence of the `role` construct in guiding the participants to answer questions correctly. In Domain 2, question Q4, we realize that GB participants indicated the correct answer (minimum 1) more than 65% of cases, while GD participants indicated 20% of incorrect answers (minimum 0) and no correct answers. The other occurrences of errors were due to other reasons. Here, we once more notice the influence of `role` to guide the correct answer, although not as significantly as in Domain 1. Regarding question Q2, the minimum 0 is the correct answer, for those who noticed the implicit condition being tested in this question. However, only 3 participants explicitly mention that they have perceived this implicit condition being tested, and thus the results of this question mask the effect of using the `role` construct. It could be interesting for us to investigate if there are situations in which the use of `role` construct can have a negative effect, as it seems to be the case here

In summary, the evidence points out that in questions involving minimum cardinalities interpretation, optional relations confuse the reader. Using `roles`, thus representing only mandatory relations, clearly leads to a better performance of the participants.

RQ2) What are the effects of the presence / absence of `relators` *in interpreting diagram fragments?*

The findings of the experiment show that in diagram fragments where `relators` are present, the tendency of the participants is to reason on them to interpret the question. Moreover, in general, `relators` contribute positively to a quality interpretation.

Table 6 gives us some interesting indications: (i) There was better performance of groups with the complete representation (total above 70%, while other representation forms did not reach 50% of correctness); (ii) When contrasting the results of GA and GC (the most experienced groups, which used the complete representation and a representation without `relator`), we also realize a better performance of each in cases when they interpreted diagram fragments that had the presence of `relators`. The result of GB also provides evidence of the positive influence of `relator`, even though less significantly; (iii) The representations with relator (first two columns of table) had better performance than representations without relator (last two columns of table). One exception occurred in the representation with relator, without role (the middle column), which seems to have been affected by the experience level of GD (there was a significant number of partially correct answers, which we count as error); (iv) We notice that a small change in the representation strategy (between columns 1 and 2, by showing the material relation derived from the relator) apparently causes a large variation in the result (correctness percentage). In this case, however, it seems to be an influence of the experience level more than of the representation strategy (GA and GC have more experience than GB and GD).

Another aspect should be highlighted: Were the participants able to notice when a question could not be satisfactorily answered by a representation? Without relator, there were attempts to follow other paths of reasoning, and usually only the most experienced participants realized the implications of the absence of the relator construct in questions where it was required. This situation should have been noted by the participants of GA and GB in the four questions of Domain 1, as well as by the participants of GC in three questions (Q2, Q3, Q4) of Domain 2. The last line of Table 5 contains some values related to it. The values are lower than expected, especially in the more experienced groups. This may be an indication that the participants tried to answer the questions with the information provided, without thinking enough about what might be missing.

3.2.5 Limitations and Validity Threats

During the experiment, we identified some limitations and validity threats. Firstly, we invited to participate in the experiment students of a course that involved different student levels (undergraduate, master and doctoral), with different experience levels in conceptual modeling. We tried to balance this diversity along the course. Thus, the participants were aware of the concepts necessary to execute the proposed activity. However, the results showed us that the different experience levels affected the experiment outcome. Secondly, we selected some unusual problem domains to be interpreted through the models, to reduce the influence of background knowledge. So, the participants could concentrate on the diagram to answer the questions. However, the use of such domains may have had an inverse effect, distracting some participants instead of allowing them to focus on the diagram. Thirdly, we considered the interpretation of small fragments of conceptual models. The understandability in larger models can lead to different results from those obtained in this experiment. This is not clear yet, as the use of `relator` and `role` constructs in these models could contribute for enlarging the models and increasing their complexity. Fourthly, the number of participants was small, and thus we had not a representative sample. Because of that, we could not apply statistical hypotheses tests. Fifthly, each group analyzed different relationship representations in different problem domains, thus analyzing different relationship types (for example, reflexive and binary relationships). The results may have been influenced by the familiarity of the participants with such aspects. Sixthly, as the number of participants was small and we were ambitious in the quantity of representation strategies that we intended to evaluate, not all strategies identified were analyzed by all groups. This fact complicated the analysis of the collected data, because not all possible variations (groups *versus* representation strategies) were available for analysis. Finally, the participants knew that the experiment aims to seek for evidence of the usefulness of `relator` and `role` constructs. The awareness of the importance of such concepts may have induced them to reason in a certain way. Indeed, as aforementioned, some participants highlighted the `relator` absence and pointed out such situation as a factor that hindered the diagram interpretation (which was expected).

3.3 Empirical Study 3: Application of PoN Approach in the Design of a DSML

There is a growing interest in Domain-Specific Modeling Language (DSML) by fields that require the elaboration and interpretation of models by non-technical stakeholders, as in Business and Administration fields. So, following this tendency there is a growing need of Modeling Language Engineering Methods. The Physics of Notations (PoN) theory has stood out, as a tool for analysis and design of DSMLs. PoN is the focus of the current empirical study.

3.3.1 Objectives

The experiment *goal* is to collect language designer impressions performing a design activity for language concrete syntax supported by the PoN theory. The *objects of study* are concrete syntaxes based on metamodels that were developed using different domain ontologies. The *research hypothesis* is that providing the language designers with guidelines[15] to perform their tasks increases the task efficacy and the designer satisfaction level.

3.3.2 Design

This experiment was conducted following the guidelines presented in (JURISTO; MORENO, 2001).

The *research strategy* was basically qualitative, as we examined not only the proposed graphical notations but also the participants impressions of using PoN to produce these notations. The *experimentation level* was in-vitro as we applied the task in a controlled environment in which the conditions were created. The *research approach* was analytical, because we were seeking to understand the behavior of participants in language design activities when they have some guidance offered by PoN.

The *subjects* were PhD and postdoctoral students possessing knowledge of conceptual modeling. This restriction reflects the fact that we believe that a visual notation design activity demands a higher maturity level, which can usually be found in conceptual modeling students of doctoral level. Also, we considered participants from two different Educational Institutions to verify possible variations according to culture and formation.

The task was divided into groups because there were two domains[16] to work with. In both groups there were participants from both Educational Institutions. There was no separation between control and experimental groups, that is, the two groups performed the same experiment task.

The *experiment task* was to propose a language concrete syntax following the guidelines given by PoN. The participants received a (UML-based) metamodel, a basic glossary complementing the metamodel and instructions based on PoN

[15] The guidelines considered in this experiment are the PoN principles.

[16] We decided to work with two domains to check if the participants impressions would be similar when they are facing different domains.

approach. As a result, following PoN principles, they should generate the symbol set for a concrete syntax (both entities and relations) corresponding to the metamodel.

The PoN theory defines nine principles. In the experiment, we decided to reduce the scope of the experiment, applying only six of those. The principles of Cognitive Fit, Cognitive Integration and Complexity Management are not considered in this experiment, because they require a different type of design that is larger and more complex – creation of different dialects associated to a same abstract syntax; larger and/or more complex domain that demands dealing with size and organization of information at different levels.

Instructions and documentation to be filled out by the participants were sent via e-mail to them. Thus, each participant could perform the task as s/he felt more comfortable with. The focus of the experiment was on the approach applied. Supporting tools used by the participants were not of a core concern at this moment. The representation choice of each concept should be totally free – a handwritten drawing, a drawing generated by a computational tool, an image extracted from the Internet. The way a language designer chooses to represent a construct or a relation is part of the indications we want to collect. For each representation that the participant designs, it was recommended that s/he identifies the design rationale adopted. This allowed us to identify some characteristics of the process adopted by individual participants.

The instruction given to the participants presented a summary of each PoN principle. Also, there were two supporting publications that were recommended to the participants as additional supporting material – the works in (MOODY, 2009) and (MOODY *et al.*, 2010).

Two problem domains were considered. Both domains were adaptations of problem domains previously used in other works. This made it possible to do some comparison between the different works (the current one and the works in which the domains were applied for the first time). The domains were similar to each other considering complexity, number of concepts and underlying OntoUML concepts that were considered. However, knowledge of UFO was not mandatory (OntoUML stereotypes were not shown in the models. The application of OntoUML was just a guidance to make the models more compatible and to offer some guidance to researchers in their observations). The domains considered were the following:

- Domain 1 (D1) – Genealogy (Figure 23). This was based on an example presented in (GUIZZARDI, 2005a). Some modifications are applied to the original model: (i) The names of Female Person and Male Person are updated to Woman and Man; (ii) Inclusion of the `relator` Parenthood to complete the representation of ParentOf `material` relation; (iii) Inclusion of the `kind` Family and two whole-part relations involving Parent and Offspring. These changes to the model were performed to make compatible the relation types presented in both domains;

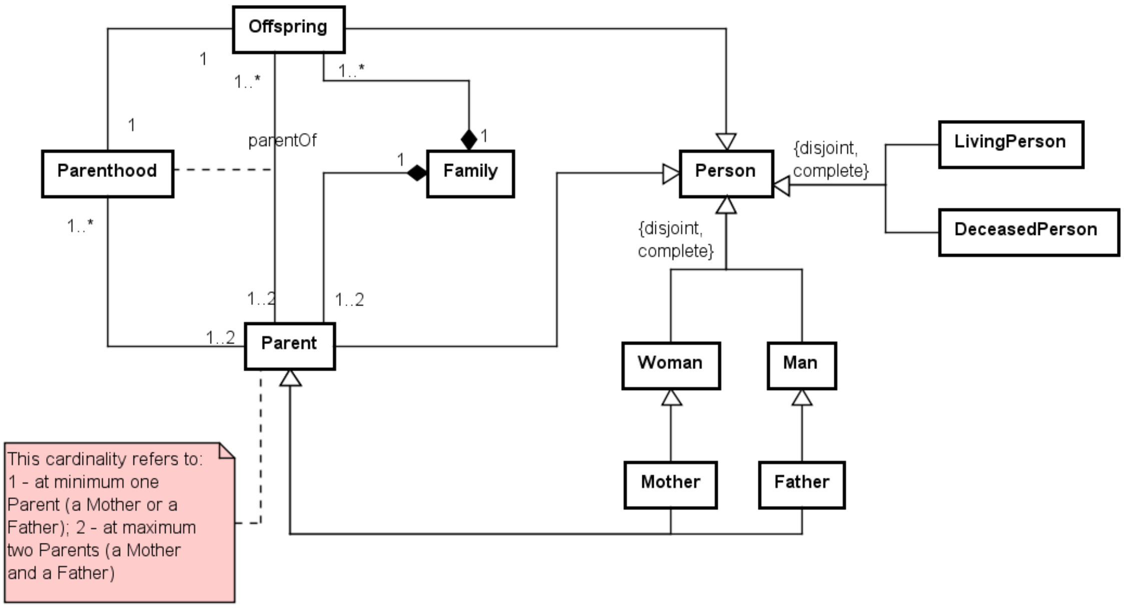


Figure 23: Language metamodel regarding domain 1 (Genealogy)

- Domain 2 (D2) – Organizational Structure (Figure 24). It is based on a DSML used in (SILVA TEIXEIRA *et al.*, 2013) (also used in the experiment described in section 3.1). Some modifications are made to the original model: (i) Elimination of Department and Organizational Branch concepts to reduce the quantity of concepts, turning compatible the number of entities between D1 and D2; (ii) As a consequence of item (i), we renamed Organizational Unit to Organization and associated this concept to Employee through a whole-part relation; (iii) Inclusion of the `relator` Subordination to complete the representation of the `material` relation reports To.

The documents filled out by the participants were:

- Profile form – This form demands information concerning the academic and professional experience of the participant;
- Language form – This form contains the metamodel and following glossary. Also, this is the place in which the participant should indicate the symbol set and adopted rationale;
- Feedback form - This form presents several questions concerning the participants impressions after performing the experiment task. The questions are related mainly to the applied approach.

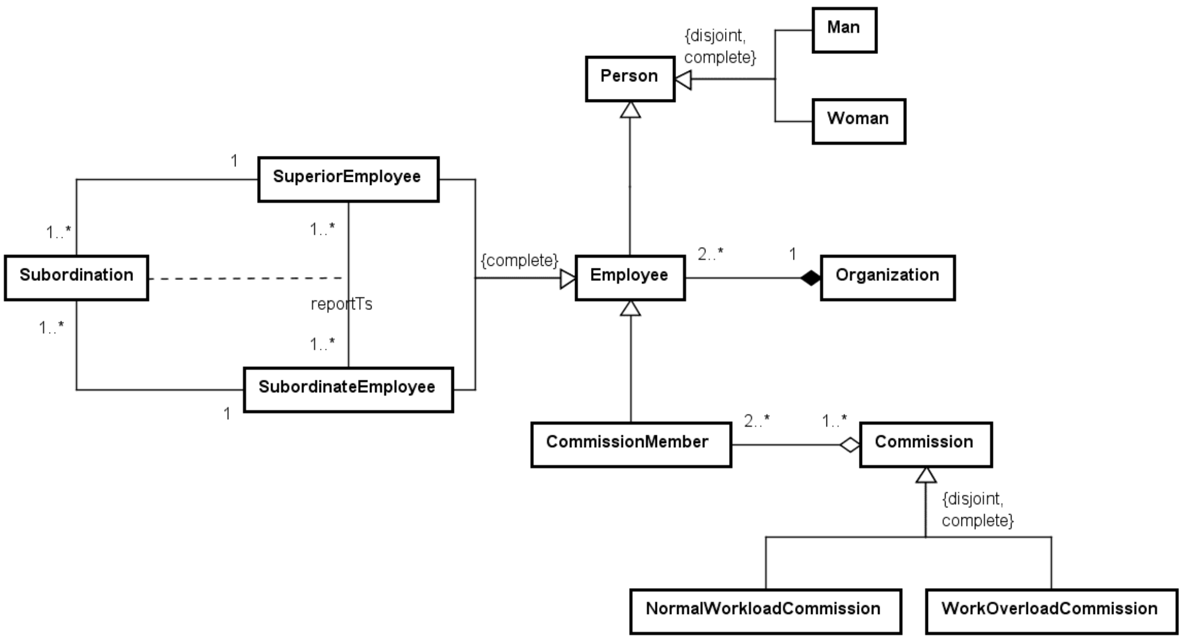


Figure 24: Language metamodel regarding domain 2 (Organizational Structure)

Complementing the information to be made available through the documents indicated above, a brief interview was scheduled with some participants. The intention was to discuss with the participants about the procedures they had adopted, to rectify missing information or to clarify possibly ambiguous information items.

3.3.3 Results

Data collected in this study was mainly qualitative, obtained from participants impressions (via language form, feedback form and interview). Basically, the data regards the resulting concrete syntax proposed by each participant and the process they had adopted to produce it. Data collected includes: ease to use (PoN approach and PoN principles), satisfaction level (PoN approach), utility (PoN principles), number of visual variables, completeness of representation (mapping between modeling elements and representational elements). These data are exposed in the tables cited next.

Table 9,Table 10 and Table 11 depict the concrete syntaxes proposed by the participants that worked with Domain 1. Table 12, Table 13 and Table 14[17] present the concrete syntaxes proposed by the participants that worked with Domain 2. Each cell presents a representational element designed by a participant to some modeling element that s/he decided that is necessary to be represented. The blank cells indicate concepts for which participants did not provide a representational element.

Table 15, Table 16 and Table 17 contain data generated from the observation of the proposed concrete syntaxes. Table 15 identifies the visual variables applied by each participant in at least one symbol. From this table, we notice that participant P4 was the participant that applied the highest number of visual variables, while participant P8 was the participant that applied the lowest number of visual variables (but even in this case, it was more than one visual variable). Table 16 and Table 17 show each modeling element that is mapped to a representational element by each participant. The listed concepts are all the elements modeled in the metamodel of each domain (Figure 23 and Figure 24). A complementary information is the column presenting a classification of entities in abstract or concrete types. From these two last tables we notice that entities were more represented than relations. In Domain 1, participant P5 represented all entities and relations, while participant P4 did not represent any relation. In Domain 2, three participants represented all the entities, while participant P8 did not represent any relation.

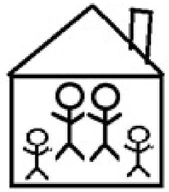
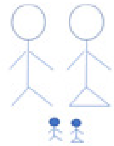
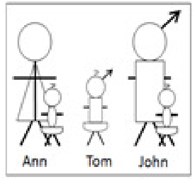
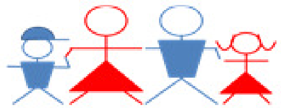
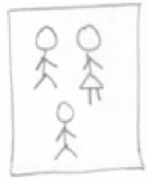
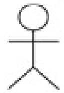
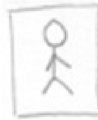
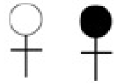
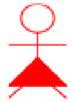
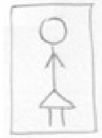
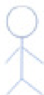
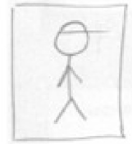
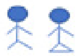
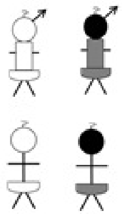
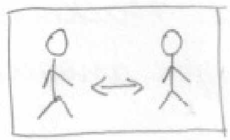
The following tables (Table 18, Table 19, Table 20 and Table 21) present data summarized from the responses given in the feedback forms. These tables represent the participants impressions in applying the six PoN principles explored in the experiment. Table 18 contains a score of utility of the PoN principles made by the participants. Each cell stores the number of participants which gave an opinion about each classification of *utility* for each PoN principle. The cells highlighted in gray on the line for each PoN principle identifies the classification of utility most voted by the

---

[17] We divided the concrete syntaxes of each domain in three tables due to visualization issues.

set of participants. Table 19 contains similar data, however, about *ease of use*. Table 20 scores the participants responses to the question: Was PoN approach easy to use? Table 21 shows a basic score of the participants satisfaction level in applying the PoN approach.

Table 9: Proposed concrete syntax for domain 1. Part 1[18]

Concept	Participant				
	P1	P2	P3	P4	P5
Family				 <<Family>>	
Person				 <<Person>>	
Woman	No symbol defined - Abstract entity			 <<Woman>>	
Man	No symbol defined - Abstract entity			 <<Man>>	
Offspring					

[18] Blank cells are concepts / relations that the participants did not indicate anything about them. This is valid in all the tables that present the proposed concrete syntaxes.

Table 10: Proposed concrete syntax for domain 1. Part 2

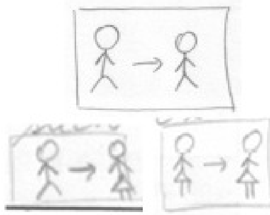
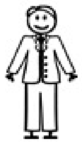
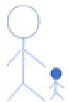
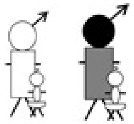
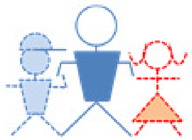
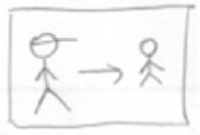
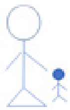
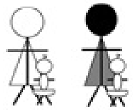
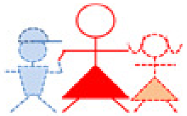
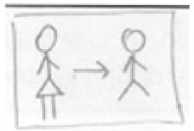
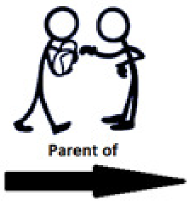
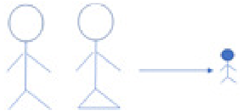
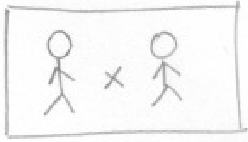
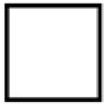
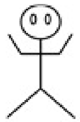
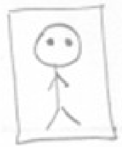
Concept	Participant				
	P1	P2	P3	P4	P5
Parent	No symbol defined - Abstract entity				
Father				 <<Father>>	
Mother				 <<Mother>>	
Parenthood			No symbol defined		
Living Person				 <<Living Person>>	

Table 11: Proposed concrete syntax for domain 1. Part 3

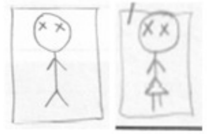
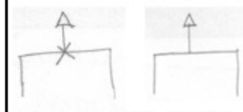
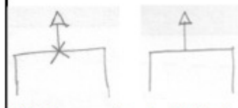
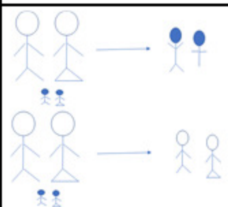
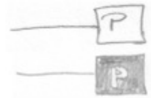
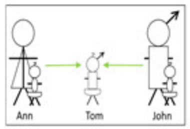
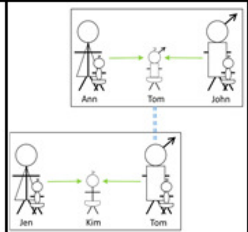
Concept	Participant				
	P1	P2	P3	P4	P5
Deceased Person				 <<Deceased Person>>	
Generalization Set 1 (Person - Alive)			No symbol defined		 {disjoint,complete} {disjoint}
Generalization Set 2 (Person - Gender)			No symbol defined		 {disjoint,complete} {disjoint}
Whole part (2 relations involving Family as whole); P1: Family Kinship					 Part of something essential part of something
ParentOf					
P3: relation between the subclasses of Parent; P5: (parenthood - parent, parenthood - offspring)					

Table 12: Proposed concrete syntax for domain 2. Part 1

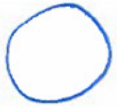
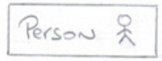
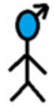
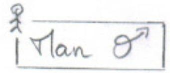
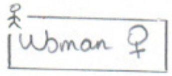
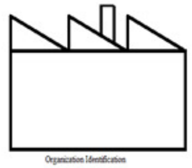
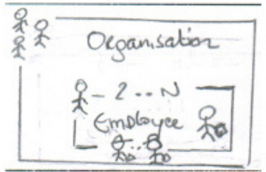
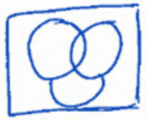
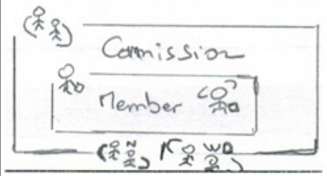
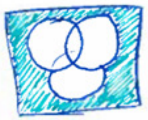
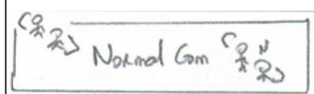
Concept	Participant					
	P6	P7	P8	P9	P10	P11
Person						
Man						
Woman						
Employee					Suggestion: "I opted to not define a specific graphical notation to "Employee". The information that a person is an employee is inferred from the fact that it has a "wholePart 1" relation with an organization. "	A composite symbol involving Employee, Generalization set Employee Subordination, Organization
Organization						
Commission						
Normal Workload Commission	 There are green smiles in the Employess images			Same symbol of Commission applying green color		

Table 13: Proposed concrete syntax for domain 2. Part 2

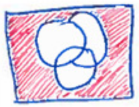
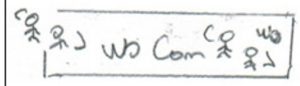
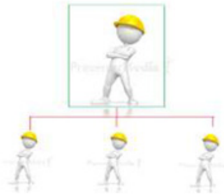
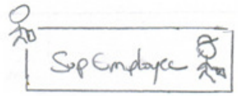
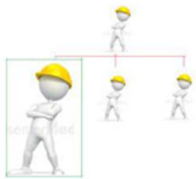
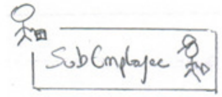
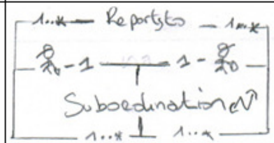
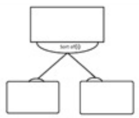
Concept	Participant					
	P6	P7	P8	P9	P10	P11
Work Overloaded Commission	 There are not smiles in the images - the mounth is red and sad			Same symbol of Commission applying red color		
Commission Member				Decision: keep the Employee representation. The meaning will be in the relation	Decision: "No specific notation was defined to comission members: an employee is a comission member if it is linked to some comission"	A composite symbol involving Commission, Commission Member, Generalization set of Commision Phases
Superior Employee					Implicity representation, through the representation of the reportsTo relation	
Subordinate Employee					Implicity representation, through the representation of the reportsTo relation	
Subordination				 same symbol of reports To	Implicity representation, through the representation of the reportsTo relation	
Generalization set 1 (Phases Commission)						A composite symbol involving Commission, Commission Member, Generalization set of Commision Phases

Table 14: Proposed concrete syntax for domain 2. Part 3

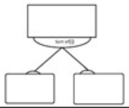
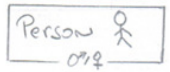
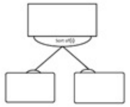
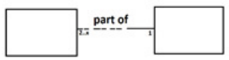
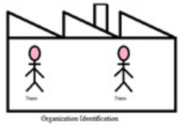
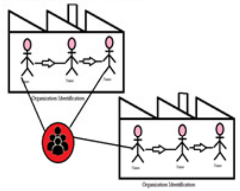
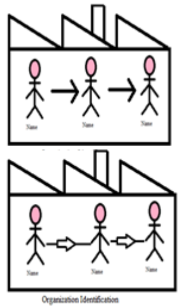
Concept	Participant					
	P6	P7	P8	P9	P10	P11
Generalization set 2 (Person Gender)						
Generalization set 3 (Employee subordination)						A composite symbol involving Employee, Generalization set Employee Subordination, Organization
WholePart relation (Organization x Employee)					 An instance example	
WholePart relation (Commission x Commission Member)					 An instance example	
reports To				 same symbol of Subordination	 two options, depending on the need of attach textual information of the relation	
Other general links						

Table 15: Application of visual variables by participants designing concrete syntaxes of Domain 1 and Domain 2

Domain	Participant	Visual Variable						
		Brightness	Color	Orientation	Position	Shape	Size	Texture
D1	P1	X	-	-	X	X	X	-
	P2	-	-	-	X	X	X	-
	P3	X	X	-	X	X	-	-
	P4	-	X	X	X	X	X	X
	P5	X	-	-	X	X	X	X
Percentage		60%	40%	20%	100%	100%	80%	40%
D2	P6	-	X	-	X	X	X	-
	P7	-	X	-	X	X	-	X
	P8	-	X	-	-	X	-	-
	P9	X	X	-	X	X	-	-
	P10	-	X	-	X	X	-	-
	P11	-	-	X	X	X	-	-
Percentage		16,67%	83,33%	16,67%	83,33%	100%	16,67%	16,67%
TOTAL	Quantity	4	7	2	10	11	5	3
	Percentage	36,36%	63,63%	18,18%	90,90%	100%	45,45%	27,27%

Table 16: Identification of concepts representation by participants that designed Domain 1 concrete syntax

Concepts		Participants				
Entity	Classification	P1	P2	P3	P4	P5
Person	Abstract	-	X	-	X	X
Man	Concrete	-	X	X	X	X
Woman	Concrete	-	X	X	X	X
Parent	Abstract	-	-	-	-	X
Mother	Concrete	X	X	X	X	X
Father	Concrete	X	X	X	X	X
Living Person	Concrete	X	X	X	X	X
Deceased Person	Concrete	X	X	X	X	X
Family	Concrete	X	X	X	X	X
Offspring	Concrete	X	X	X	-	X
Parenthood	Concrete	X	X	-	-	X
Percentage (%)		63,63	90,90	72,72	72,72	100
Relation						
Generalization Set 1 (Person - Alive)		-	X	-	-	X
Generalization Set 2 (Person - Gender)		-	X	-	-	X
Parent is Person		-	-	-	-	X
Mother is Parent		-	-	-	-	X
Father is Parent		-	-	-	-	X
Offspring is Person		-	-	-	-	X
Whole-part - Family and Offspring		X	X	-	-	X
Whole-part - Family and Parent		X	X	-	-	X
ParentOf		-	X	X	-	X
Parenthood + Offspring + Parent		-	-	-	-	X
Percentage (%)		20	50	10	0	100
Total Percentage (%)		42,85	71,42	42,85	38	100

Table 17: Identification of concepts representation by participants that designed Domain 2 concrete syntax

Concepts		Participants					
Entity	**Classification**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**
Person	Abstract	X	X	X	X	X	X
Man	Concrete	X	X	X	X	X	X
Woman	Concrete	X	X	X	X	X	X
Employee	Concrete	X	X	X	X	-	X
Organization	Concrete	X	X	X	X	X	X
Commision	Abstract	X	X	X	X	X	X
Normal Workload Commission	Concrete	X	X	X	X	X	X
Work Overloaded Commission	Concrete	X	X	X	X	X	X
Commision Member	Concrete	X	X	X	-	-	X
Superior Employee	Concrete	X	X	X	-	-	X
Subordinate Employee	Concrete	X	X	X	-	-	X
Subordination	Concrete	X	-	X	X	-	X
Percentage (%)		**100**	**91,66**	**100**	**75**	**66,66**	**100**
Relation							
Generalization set 1 (Phases Commission)		X	X	-	-	-	-
Generalization set 2 (Person Gender)		X	X	-	-	-	X
Generalization set 3 (Employee subordination)		X	X	-	-	-	X
Whole-part (Organization x Employee)		X	X	-	X	X	-
Whole-part (Commission x Commission member)		-	-	-	X	X	-
reports To		X	X	-	X	X	-
Employee is Person		-	-	-	-	-	-
Commision Member is Employee		-	-	-	-	-	-
Subordination + Superior Employee + Subordinate Employee		-	-	-	-	-	-
Percentage (%)		**55,55**	**55,55**	**0**	**33,33**	**33,33**	**22,22**
Total Percentage (%)		**80,95**	**76,19**	**57,14**	**57,14**	**52,38**	**66,66**

Table 18: Results for question "Considering the utility (how useful it is) of each PoN principle, identify a score of utility for each principle"

PoN Principle	Type of Answer (Quantity of Participants)				
	Completely useless	**Useless**	**Indifferent**	**Useful**	**Completely Useful**
Semiotic Clarity	1	2	1	3	**4**
Perceptual Discriminability	1	1	0	**5**	4
Semantic Transparency	1	1	0	3	**6**
Visual Expressiveness	1	1	0	**5**	4
Graphic Economy	0	2	2	0	**7**
Dual Coding	0	3	**4**	3	1

Table 19: Results for question "Considering the ease to use (how easy it is to use) of each PoN principle, identify a score of ease to use for each principle"

PoN Principle	Type of Answer (Quantity of Participants)				
	Very difficult	Difficult	Indifferent	Easy	Very easy
Semiotic Clarity	2	2	1	**5**	1
Perceptual Discriminability	0	**3**	**3**	2	**3**
Semantic Transparency	1	**3**	2	**3**	2
Visual Expressiveness	2	**3**	**3**	**3**	0
Graphic Economy	0	**3**	**3**	**3**	2
Dual Coding	0	0	3	**6**	2

Table 20: Results for question: "Was PoN approach easy to use?"

Type of Answer	Quantity of Participants
Very difficult	0
Difficult	**5**
Indifferent	1
Easy	**5**
Very easy	0

Table 21: Results for question: "What is your satisfaction level concerning the approach you applied?"

Type of Answer	Quantity of Participants
Completely unsatisfied	0
Unsatisfied	0
Indifferent	2
Satisfied	**8**
Completely satisfied	1

The main impressions made by the researchers from such tables are indicated in the following section.

A questionnaire was applied to capture the participants profile. Regarding the sample size, there were 11 participants (a small sample size). The participants profile is of high maturity students (PhD and postdoctoral level) acting as language designers with knowledge in Conceptual Modeling (usually in UML). Some of the observation we can make after analyzing the questionnaires include:

- Experience and knowledge of the participants are divided among Computer Science, Applied Economics, Business Informatics and Management Information Systems fields;

- In general, the participants have UML knowledge. Only four participants indicated less than 1 year of experience in that. Their experience is higher in language use (model development and interpretation) than in language development (design and evaluation tasks);
- All participants have some experience level in modeling languages other than UML (BPMN, ER and Archimate being the most common). As for the case of UML, the most common task in which these participants have experience on is model development and interpretation;
- Concerning the application of PoN, most participants did not have previous knowledge of the framework. Among the participants that have had previous knowledge of PoN, just one had previous experience applying it;
- Concerning knowledge in a different design approach other than PoN, only one participant indicated that s/he has applied a different approach to language design. However, s/he highlighted that it was an abstract syntax design approach, while referring to UFO[19].

3.3.4 Discussion

Based on the data collected, we formulated some indications that are exposed in the sequel.

- In general, the proposed symbols are quite similar among the different proposals. In any case, no two proposals are completely alike.

[19] The participant answered simply "UFO". We highlight that UFO is not a design approach, but it can be included in a design approach as occurs with the ontological guidelines presented in (GUIZZARDI, 2013).

Concerning Domain 1, the proposals keep the similarity with the original proposal made in (GUIZZARDI, 2005a)[20]. Probably, this occurs because the domain is quite simple and well known.

In Domain 2 there are more differences than in Domain 1. For example, entities Employee and Organization have different symbols proposed by each participant, even if there are similarities. A possible explanation for this effect is perhaps because familiar concepts such as Person and Family have a more direct shared representation on people mind than concepts such as Employee, Organization and Commission;

- Usually, the symbols in a same proposal are quite similar and we can notice the connection between them. This can be consequence of the Perceptual Discriminability principle application;
- Composite symbols (addition of details) seems to be the most common choice to represent the concepts;
- Usually, the symbols proposed are clearly connected to the domain. This can be a consequence of the Semantic Transparency principle application;
- Concerning representation of abstract types: (i) Some participants decided not to represent abstract symbols, recognizing that it can be unnecessary to do so. For example, the entity Parent was represented only by participant P5. This can be a consequence of the Graphic Economy principle application; (ii) Participant P10 justified her/his decision of representing Person, commenting that depending on the situation it would not be necessary to identify a person gender. So, we have overall a situation in which sometimes (i) Abstract types are not represented consciously and sometimes (ii) Abstract types are represented consciously;
- Apparently, it is easier for the participants to represent entities than relations. An indication of this is the frequency of blank cells in the lines corresponding

[20] See Figure 2-18 in (GUIZZARDI, 2005a). In that moment, Guizzardi didn´t know PoN, published for the first time in 2009, nevertheless the symbols chosen for him kept the same idea of the new proposals. This can be an indication that common sense is highly applied when choosing symbols to represent a concept and that PoN principles reinforce this.

to the relation concepts. For example, participants P4 and P8 did not represent any relation directly. Also, usually entities representation is more connected to the domain (use of symbols that reflects the concept meaning) than relation representation, in which the participants apply tradition in modeling language (as the UML symbols), that is, abstract symbols (not directly connected to the concept meaning);

- The theory of Signs (PEIRCE; BUCHLER, 1940) establishes that a symbol can be an icon, an index or an abstract symbol. From the observation of the proposed concrete syntaxes, we identified that index is the most common type of symbol applied, over signs and abstract symbols;
- Some participants are aware of symbol anomaly. For example, participant P1 indicated consciously the same symbol to the whole-part relations of Family;
- There were cases of redundancy. For example, participant P10 applied color and adding of detail in the shape to represent Man and Woman. We could not verify why such decision was made;
- Some participants were concerned about how the defined symbols could be combined. For example, the participant P9 when defining a symbol for Employee commented that representated it through adding of detail to the Person symbol, and that the solution can be repeated in the symbols representing Man and Woman when necessary;
- Concerning Offspring, a subtype of Person, it is not directly connected to Man and Woman (also subtypes of Person) as Father and Mother entities. Some participants chose to represent an Offspring without considering its connection to Man and Woman (as P1 and P5). On the other hand, participants P2 and P3 considered this connection and created specific symbols to Offspring-Man and Offspring-Woman. It is unclear which the best option is. It depends on the needs of the modeling task, namely, whether it is necessary to identify if an offspring is male or female;
- In general, our expectation was that having some guidance to follow and the need to describe the design rationale would guide the participants in a more in-depth reasoning process. Some decisions appear to be taken consciously. For example, P1 decided not to represent abstract entities and justified the

decision claiming that this type is not instantiated. On the other hand, some participants seem to be unaware of their decision. An example is P5 that simply proposed a symbol for each model element, including a rectangle in each symbol. Both these situations can be taken as indication that PoN guidance provided some help, but used in isolation it is not enough to fully support language designers;

- Participant P3 applied examples to show several decisions. Even a new situation is demonstrated through examples (that a same Person can be involved in more than one Family instance in different roles). This can be an indication that examples could be a good support in the design process;
- The choices of participant P11 had some particularities: (i) S/he worked with decorative symbols; (ii) S/he recurred strongly to textual support. This could be an indication that s/he has the profile of a textual person[21]. Unfortunately, s/he did not write down the design rationale used. During the interview, s/he commented that s/he chose to work in this way (that is, by using both graphic and text) to reinforce memorization of the represented elements. This case induced us to think that letting the designers to explicitly express the design rationale adopted to define a symbol can be helpful for enabling understanding of the symbol itself;
- Participant P9 used the metamodel characterization to explain some decisions, even if this was not explicit in the model. So, it is a case in which the participant referred to ontological properties. S/he indicated that it would not be necessary to represent Superior Employee and Subordinate Employees (they are `roles`), given that their representation could be inferred through the representation of the reports To relation (the `material` relation). Also, s/he decided to represent Subordination (a `relator`) and reports To relation using a single symbol, which is similar to the guideline suggested in (GUIZZARDI, 2013). The consideration of ontological properties is also clear

[21]As highlighted by Moody (MOODY, 2009): “people differ widely in their spatial and verbal processing abilities. Including graphics and text, it is likely to improve understanding by people across the full spectrum of spatial and verbal abilities”. Here, what we called as textual person is a person whose verbal abilities are higher then his/her spatial abilities.

in a registered comment: “The instances of Person could assume different roles (at the same time). Thus, I think the visual notation should deal with that. For example, the symbols for man and woman should be compatible with employee, in a way that it is possible to represent man employer and woman employee (orthogonality)”;

- Participant P10 explained why s/he decided to represent the two whole-part relations differently. Basically, s/he identified that an Employee can be associated to only one Organization, but it can be part of several Commissions. Again, this is an indication that participants frequently employ their knowledge of ontological properties when reasoning about aspects of language design;
- In general, the participants explained their design rationale, in greater or lesser detail. However, we notice that few participants have actually resorted to using the PoN principles in such justifications (at least they did not explicitly quote them in their justifications). We consider this as an indication that we need to reinforce the explicit use of such principles;
- The participants do not follow the same sequence for choosing the modeling elements to represent their symbols (they were not asked to do so), but the sequence of the drawings disposed in the language form is an indication in this case. What is clear is that the participants usually start in the top of a taxonomy, and how they transverse such a taxonomy varies. Also, entities are usually fully represented before addressing the representation of relations. Another indication is that participants usually studied the metamodel before they move to proposing the notation. So, they are aware of the whole model when establishing how to represent a symbol that can be affected by another. For example, participant P10, when establishing the representation for Organization, indicated that s/he made a drawing that later allows Employee to be attached to it, as required by the whole-part relationship between them;
- Participants with some knowledge of ontologies (in this case, UFO) ended up explaining their decisions using ontological properties, even if they are not clearly presented in the metamodel. This was the case, for example, with participants P9 and P10.

According to Table 15: (i) In Domain 1, shape and position were the most applied visual variables. Only one participant applied orientation (P4, to represent the phases Living Person and Deceased Person); (ii) In Domain 2, shape was the most applied visual variables. On the other hand, brightness, orientation, size and texture were the least applied visual variables; (iii) We can notice that, except shape and orientation, the choice of which visual variables to apply varied between the groups; (iv) In general, the use of visual variables varied more in group 1. It is difficult to say conclusively whether these differences in the result are caused by the participants profile (quite similar between the groups according to the analysis of the participants profile. Thus, it is less likely to be the cause of differences) or by the domain (the probable cause of the differences).

According to Table 16 and Table 17, in Domain 2 more concepts (entities + relations) were represented. However, it was in Domain 1 that only one participant represented all concepts. As in the previous item, we deduce that this difference was caused by the domain characteristics instead of by the participants' profiles.

Analysis of Table 18 shows that PoN principles, in general, are considered helpful. But this was not a unanimous opinion. Dual Coding is the less useful in the participants opinion. On the other hand, Graphic Economy and Semantic Transparency are the most useful principles. Justification about these scores vary from "The designer need to choose which are the most important in each situation" to "the concepts I marked as 'Completely useful' will lead to better models. The others merely to a better or quicker understanding by novices." This is compatible with the idea that PoN principles should be adopted in different levels according to the requirements of the language.

Analysis of Table 19 and Table 20, are about the "easy of use" of PoN principles. As these tables show, opinions are divided in this issue. Here are some participant justifications when considering PoN in that respect: (i) "I think that the judgment of these principles are, in general, very subjective. For sure the guidelines proposed by the author help a lot, but I cannot see how to judge the achievement to the principles objectively. Even the "semiotic clarity" which in a first view seems to be very objective, in practice it was shown to be somehow imprecise"; (ii) "The principles were clear to me, and most of them are very intuitive. It was still not easy to choose element to represent the entities that form the metamodel". These are indication that the PoN process should be improved.

In any case, even if the participants pointed several problems and difficulties in applying PoN, most of them indicated to be satisfied regarding the approach, as evidenced in Table 21.

The participants were also asked about suggestions they make to improve the application of PoN. Some suggestions were:

- "A sort of step-by-step guide would be very helpful.";
- "A table with questions (check list)";
- "Maybe something shorter, like how to design just a supertype and generalisation sets";
- A participant commented that the problem is not to apply each principle separately, but try to apply them together (pointing that an approach that considers the relation between the principles and their application in tandem should be considered).

3.3.5 Limitations and Validity Threats

During the experiment, we identified some limitations and validity threats.

- Listed threat: Heterogeneity of participants. There may be a wider range of profiles and experience than we considered previously. The profile form will guide this analysis and if this possibility becomes true, we will try to get around it (creating more groups and analyzing them separately).

  Analysis after the empirical study execution: Obviously there are differences between the participants profile, as they have different academic fields and belongs to different research groups. However, profiles are relatively balanced between the two research groups involved. Even more, we assured that participants of these two profile groups analyzed both Domains 1 and 2, reducing the impact of the different profiles;

- Listed threat: Heterogeneity of the problem domains addressed. We are addressing two domains that we judge similar in size and complexity. We are planning to evaluate their evidences together. However, if we notice some interesting indication that separate both domains we can conduce the analysis separately, identifying what caused this division.

  Analysis after the empirical study execution: As identified during the discussion, we can notice some difference among the results of each domain. Participants that worked with Domain 1 applied more visual variables. Participants that worked with Domain 2 represented more concepts. Unfortunately, we do not have enough data to analyze these differences. It could be necessary to apply another empirical study, focusing on this issue;

- Listed threat: The instructions developed to support the task execution may not be enough, helping participants only partially.

  Analysis after the empirical study execution: An important fact that we would like to analyze was the design rationale of the participants, in particular the use of the PoN principles. However, few participants identified the PoN principles used. Probably because this was not clearly requested in the instructions;

- Listed limitation: We adopted simple and small problem domains. They are simple to represent and interpret. A generalization of the results obtained with the two problem domains can be done only if we maintain these characteristics (similar size and complexity). For larger and / or more complex problem domains, we cannot guarantee the generalization of the results. Another experiment should be applied in this case.

3.4 Final Considerations

A summary of the main characteristics of the exploratory empirical studies is exposed in Table 22. The table is helpful to differentiate among the studies and to visualize them as part of the same intention, that is, to form an in-depth comprehension of the DSML concrete syntax design aspects.

As these studies were previous to a clear definition of the direction we would adopt in our research they have significant different characteristics, besides their main goals. They covered different levels (language and sentence), different activities (use and development of language), dealit with different diagrams (type elements and instance elements).

Table 22: Summary of the main characteristics of the exploratory empirical studies

	Research			Task	
Experiment	**Goal**	**Focus**	**Main Result**	**DSML level**	**Activity**
Section 3.1	To identify if interpretation of diagram instances built upon an UFO-based guidelines is of high quality	UFO-based ontological guidelines	Ontological guidelines help in the design of DSML concrete syntax. But in the current format, they are not enough	Sentence	Interpretation of instance elements diagram instances (language use)
Section 3.2	To identify if different representations of a same relation conducts to different results	Relations in conceptual modeling	`role` and `relators` help in interpreting relations. So, it is interesting to offer flexibility and include the representations of these constructs	Sentence	Interpretation of type elements diagram (language use)
Section 3.3	To collect impressions of novice designers when applying PoN in design tasks	PoN	PoN helps in the design of DSML concrete syntax. But in the current format, it is not enough	Language	Design of language (language development)

We developed studies of language use and language development. Concerning use of language, only the reading of model tasks was evaluated – even if considering different elements (type and instances). Tasks of writing models were not analyzed in our research. We considered the collected data enough to elaborate our proposals. Anyway, as future research, we intend to invest more on this.

Experiments 1 and 3 had a clear connection with the artifacts that we developed later, giving us a higher security in the guidelines we adopted (the ontological guidelines in the case of UFO, the information visualization guidelines in the case of PoN). The experiment 2 had a smaller influence, being an analysis of the importance of the use of `roles` and `relators` in the interpretation of diagrams when dealing with the representation of relationships, an issue of our interest and that aided in the establishement of a new ontological guideline.

We are conscious that the studies performed did not provide us with a complete set of evidences, a set able to show us all the variations and possibilities involving the use and the development of a DSML. However, they were enough to show us the main directions of our research. Each of the experiment gave the researchers several indications that we would then considered in our work proposing an extension of PoN and the UFO-based guidelines. We list some of these issues in the sequel:

- The establishment of design questions could be helpful to language designers, as pointed out by a participant of empirical study 3 – "A table with questions (check list)";
- The creation of a sequence of activities can facilitate applying PoN in language design. These activities should address needs that include the following: flexibilization of decisions, evaluation of proposals, detailing of how to apply a principle / activity - as pointed out by a participant of empirical study 3 – "A sort of step-by-step guide would be very helpful.";
- Concerning flexibilization of decisions and how difficult it is to represent relations, it could be interesting to allow different alternatives for their representation;
- Concerning evaluation of proposals, an activity to generate diagrams can enable the designers to observe the results of their notation proposals;
- Concerning additional information that could inform designers of how to apply a principle / activity, the experiments showed evidence that indicate the possible adequacy of employing ontological theories for this purpose;
- Use of ontological (UFO-based) and visualization of information (PoN-based) guidelines can give rise to the necessary detail level that a design process for VML concrete syntax demands;
- Experiment 2 showed us that at the type level the use of `relators` and `roles` stands out for several reasons: (i) They are recurrent; (ii) They were

not explored in the original ontological guidelines; (iii) When drawing type elements, they help to interpret relationships and we induced that the same should occur for dealing with instance elements. So, the ontological guidelines could be expended to deal with `role` and `relator` constructs.

Also, we highlight a shortcoming in the three experiments: they were applied in simplified domains (or domain fragments). It would be interesting to apply the same type of studies in larger and more complex domains for analyzing if the collected evidences would be similar.

Chapter 4. Physics of Notations Systematized (PoN-S)

Although PoN has been applied in the analysis of several existing VMLs, its application is not easy (VAN DER LINDEN *et al.*, 2016). Firstly, this is because PoN does not prescribe any method or process for systematically applying its principles (STORRLE; FISH, 2013)(GENON *et al.*, 2010). Additionally, PoN is usually more applicable to analysis activities than to design activities. As discussed in Section 2.2, some works have added elements to the basic theory intending to enable PoN for the design process. However, there are still problems. When a designer is applying PoN principles in a design process, s/he needs more design guidance. For instance, when should s/he apply a particular principle? In which sequence should principles be applied? Which principles should be applied in tandem?

To solve this issue, we have proposed a systematized approach for applying PoN in the design of DSML concrete syntaxes, called ***PoN-Systematized (PoN-S[22])*** (SILVA TEIXEIRA *et al.*, 2016). PoN-S defines a process comprising an ordered set of activities and suggestions of when to apply PoN principles. Also, PoN-S establishes a way of grouping the principles and basic design questions that should be answered by a designer. These proposals are solutions to the limitation (i) presented in Section 2.4 – As PoN does not contain a design process, we include a process model and details to each design activity.

This chapter is structured as follows: The PoN-S approach comprises design questions (presented in Section 4.1), groups of PoN principles (described in Section 4.2) and a design process (identified in Section 4.3). In Section 4.4 we present a case study that applied PoN-S. Finally, Section 4.5 identifies some final considerations related to the PoN-S approach.

---

[22] Pons is a region of the brainstem with neural pathways that carry sensory signals including those related to eye movement. Etymologically, the term from Latin also means “bridge”. Here, we believe that the systematization of PoN helps to bridge its theoretical postulations and its use in practice.

For helping to understand PoN-S, in Table 23, we present the terminology used to describe PoN-S.

Table 23: Basic terminology of PoN-S

Term	Definition
Activity	A step to be executed when performing the design process.
Principle	A recommendation to be followed. It refers to PoN principles, as identified in 2.2. A principle is an informational visualization guideline in the design processo.
Model / modeling element	Construct that composes a modeling language. It can be an entity or a relation.
Representational element (symbol, sign)	A visual representation (symbol) used to represent a model element. It can be graphical or graphical combined with textual elements.
Composite symbol	A symbol composed of more than one shape.
Language abstract syntax / VML abstract syntax	The set of modeling elements (constructs) that compose a language metamodel.
Language concrete syntax / VML concrete syntax	The set of representational elements of a VML.
Language (type) level	Refers to the elements of a language, such as its syntax, semantics and rules.
Sentence (instance) level	Concerns the application of a language, that is, building diagrams using elements defined in the language level.
Dialect	A specific set of symbols defining a concrete syntax with a specific purpose, given by a set of requirements (for example, modeling task, stakeholder profile, problem domain characteristics). For example, a dialect can be designed for novice profile, another for users possessing more graphical skills, and so on.
Language dialect set / VML dialect set	The set of dialects required for a VML. In this case, the language concrete syntax comprises the dialect set.
Dialect goal	Identification of the dialect purpose. A desired result of language application in a modeling task. For example, helping novice language user in interpretation tasks.
Dialect directives	Ways to achieve the dialect goal, principles that should be respected. For example, reinforce the Perceptual Discriminability principle.
Modeling task	A task in which a model is applied. Basically, there are two types of tasks in which a model can be used: development (codification) and interpretation (decodification).
Stakeholder profile	Main characteristics of the intended language users. For example, non-technical users.
Problem domain characteristics	Characteristics of the problem domain in which the language will be applied that influences its design. For example, structural, processual,

	simple / complex, small / large.
Representation strategy for managing model complexity	A strategy suggesting a way to organize the visualization of the representational elements of a language.
Redundant coding	Refers to a symbol that has two or more visual variables for uniquely identifying its meaning.
Composite coding	Refers to a symbol that requires two or more visual variables for uniquely identifying it.

4.1 Design Questions

When designing a VML concrete syntax, we should deal with concerns at different levels. First, we need to decide whether different dialects for the same abstract syntax are needed. The reasons for creating more than one dialect should be clearly identified (for example, the fact that the language must be suitable for more than one stakeholder profile, modeling task or problem domain characteristics). Second, at language level, we need to determine the symbols composing the concrete syntax for each dialect. Finally, at sentence level, we should be concerned with development of diagrams using the proposed concrete syntaxes. Table 24 presents the concerns for these different levels as design questions and identifies the PoN principles that can be applied to answer them.

Table 24: Answering to some basic design questions with PoN principles

Design Question	Related PoN Principles
Dialect set	
Do we need different dialects for representing the abstract syntax? If so, which dialects should we consider? What are the reasons to define each dialect?	Cognitive Fit
For each dialect	
Language level	
Which symbols do we need to create?	Semiotic Clarity
How to create each symbol?	Semantic Transparency
How to relate differents symbols? To what extent two or more symbols should be similar / different?	Perceptual Discriminability
How visual variables (such as shape, color and texture) and text should be applied in order to aid the identification of each representational element?	Visual Expressiveness Graphic Economy Dual Coding
Sentence level	

Which procedures should we develop to support managing complexity in diagrams? (Depending on the answer to this question, it may be necessary to create new symbols, affecting decisions at language level).	Complexity Management Cognitive Integration

Answering these design questions helps a designer to understand the rationale behind the application of each principle, acting as an initial guide. However, this is not enough for completely guiding the design effort. To do so, we define a way of grouping PoN principles and a process for applying them when designing a concrete syntax, which are discussed in the next two sections, respectively.

4.2 Groups of PoN Principles

Moody describes a number of influence relations between pairs of principles (MOODY, 2009). However, often these principles act in larger groups. This perception is fundamental to guide the VML design process. Thus, we suggest grouping PoN principles into four groups, described below. The principles in a group can interact with principles of another group. Furthermore, principles inside a group interact with each other. Typically, this intra-group relationship is stronger than inter-group relationships. This is the main reason for grouping the principles in such way.

Group 1 – Basic principles

This group comprises three principles: Semiotic Clarity, Semantic Transparency and Perceptual Discriminability. These principles are considered basic principles, because they should be applied to some extent in the design of any concrete syntax. They are complementary in the sense that we need to create a symbol for each construct (Semiotic Clarity), each symbol should be clearly identifiable (Semantic Transparency), and clearly distinguishable from other symbols (Perceptual Discriminability). So, these principles should be applied together in the design of each dialect.

The level at which they must be attended may vary in each dialect. Semiotic Clarity acts as a guarantee that the mapping between abstract and concrete syntaxes is complete, avoiding possible anomalies, that is, assuring that all necessary symbols are defined. Perceptual Discriminability is concerned with whether such symbols are adequately different from (or similar to, depending on the case) the others. Finally, Semantic Transparency is concerned with whether each symbol has its meaning easily inferred by the users of a particular dialect.

Group 2 – Information complexity management principles

This group comprises two principles: Complexity Management and Cognitive Integration. These principles are commonly applied when dealing with large or complex diagrams. They are complementary, since the former deals with how to organize the information in a model (probably separating them in several diagrams), and the latter refers to how to keep connection and traceability of the information spread in different diagrams. Thus, they should be applied together. Basically, this group of principles will be applied at sentence level, giving rise to representation strategies for managing model complexity. Ideally, the way of addressing information complexity management should be the same (or very similar) in any dialect. Finally, it is worth pointing out that the application of these two principles can demand creation of new symbols, hence, affecting the language level in which they are applied (Group 1).

Group 3 – Supporting principles

The supporting principles are: Visual Expressiveness, Graphic Economy and Dual Coding. Visual Expressiveness is connected to the other PoN principles (except Dual Coding), in the sense that it provides mechanisms (as visual variables) for implementing the other principles. Also, Graphic Economy is connected to the other principles, since it establishes a way to control them, trying to keep the symbols as simple as possible. We consider Dual Coding to refer only to redundant textual representational. So, it provides textual support to graphical representational elements, when both representations can be useful to achieve a better interpretation of the symbols.

Since the supporting principles have a connection with the other PoN principles, they can somehow affect principles of groups 1 and 2.

Group 4 – Dialect set principle

This group, in fact, contains only one principle: Cognitive Fit. This principle has an indirect connection to the other PoN principles, because other principles are applied to each dialect of the concrete syntax at a time, while Cognitive Fit is about defining the set of dialects.

4.3 Design Process

The design questions and grouping of principles described give the language designer important guidance for designing the concrete syntax. However, to truly systematize the application of PoN, we need a *design process* for guiding this task.

In PoN-S, the VML design process starts with concerns related to the dialect set definition. In sequel, concerns related to language and sentence levels for each dialect are worked out. The design process finishes with an evaluation of each dialect, when the dialects can be employed by language users. During its regular use, improvement opportunities can be identified and PoN-S reapplied.

Figure 25, Figure 26, Figure 27, Figure 28, Figure 29, Figure 30, Figure 31 present PoN-S design process. Each figure presents part of the process, including their inputs, outputs[23], activities and decisions to be made. The process is represented by means of an extension to UML activity diagrams[24], introducing new modeling elements, namely: guidelines and groups of guidelines (corresponding to PoN principles and PoN principles groups, respectively). A PoN principle can be

[23] Only major inputs and outputs are clearly identified in the diagrams. Secondary, intermediary or indirect results are omitted. These include, for example: a model element to be represented, results of analyses and evaluation tasks, evaluation report. They are not represented in diagrams, only cited in text (see Appendix B). The intention is to reduce visual clutter.

[24] We chose representing PoN-S design process using UML activity diagram, because it is one of the most applied diagram to describe process models (LANGER *et al.*, 2014). So, it is a notation with which the target audience of PoN-S is probably acquainted. Also, Moody and Van Hillegersberg (2009) claims in favor of this UML diagram: "*Use case diagrams and activity diagrams are the best from a visual representation viewpoint, which may explain why they are the most commonly used for communicating with business stakeholders*".

seen as an information visualization guideline to perform an activity and it is represented by means of an ellipse, which is connected by an arrow to the activity where it applies. For representing PoN groups of principles we use a similar representation but with dashed lines. Groups contain in their interior the principles that are part of the group.

The text that goes along with the diagrams identifies the purpose of each activity. Appendix B presents a more detailed description of the activities, identifying what is expected to be done in each one and the connection to the PoN principle(s).

Figure 25 depicts an overview of the VML concrete syntax design process, which comprises the following phases: *Specify dialect(s), Implement dialect(s), Validate dialect(s)*. These phases compose the design cycle. In the *Specify dialect(s)* phase, the designer shall define the dialect set for the VML, its requirements, goals and directives. In the *Implement dialect(s)* phase, each dialect should have its symbol set and representation strategies developed. Next, in the *Validate dialect(s)* phase, each dialect should be evaluated not only by the language designer but also by potential language users, whenever possible. Once the language has been evaluated, if it is considered suitable for application, it becomes available for use. During its use, some problems may be detected, requiring reapplying the whole design process.

As inputs for the design process as a whole, there are: the language abstract syntax and model instance(s). Concerning the language abstract syntax, it should be highlighed that this input includes two components: (i) A graphical metamodel identifying the language constructs and relations between them; (ii) A glossary of terms as a textual support detailing the language terms. Language dialect set, concrete syntax, and representation strategies for managing model complexity are the main outputs of the design process. Furthermore, Figure 25 presents the PoN principles applied in each phase, including their groups.

Figure 26 presents the *Specify dialect(s)* phase. In this phase, first the designer shall identify the requirements for the VML (modeling task, stakeholder profile, problem domain characteristcs) that help defining the number of dialects required. Next, in the activities *Identify dialect goal* and *Identify dialect directives*, each dialect should be characterized, establishing its goal and directives for its design. In these activities, the designer should consider the influence relations (conflicts or synergies) that exist among PoN principles (see Section 2.2). Usually, it is not possible to establish the same level of compliance to all principles. So, the designer should choose the principles to highlight in each dialect.

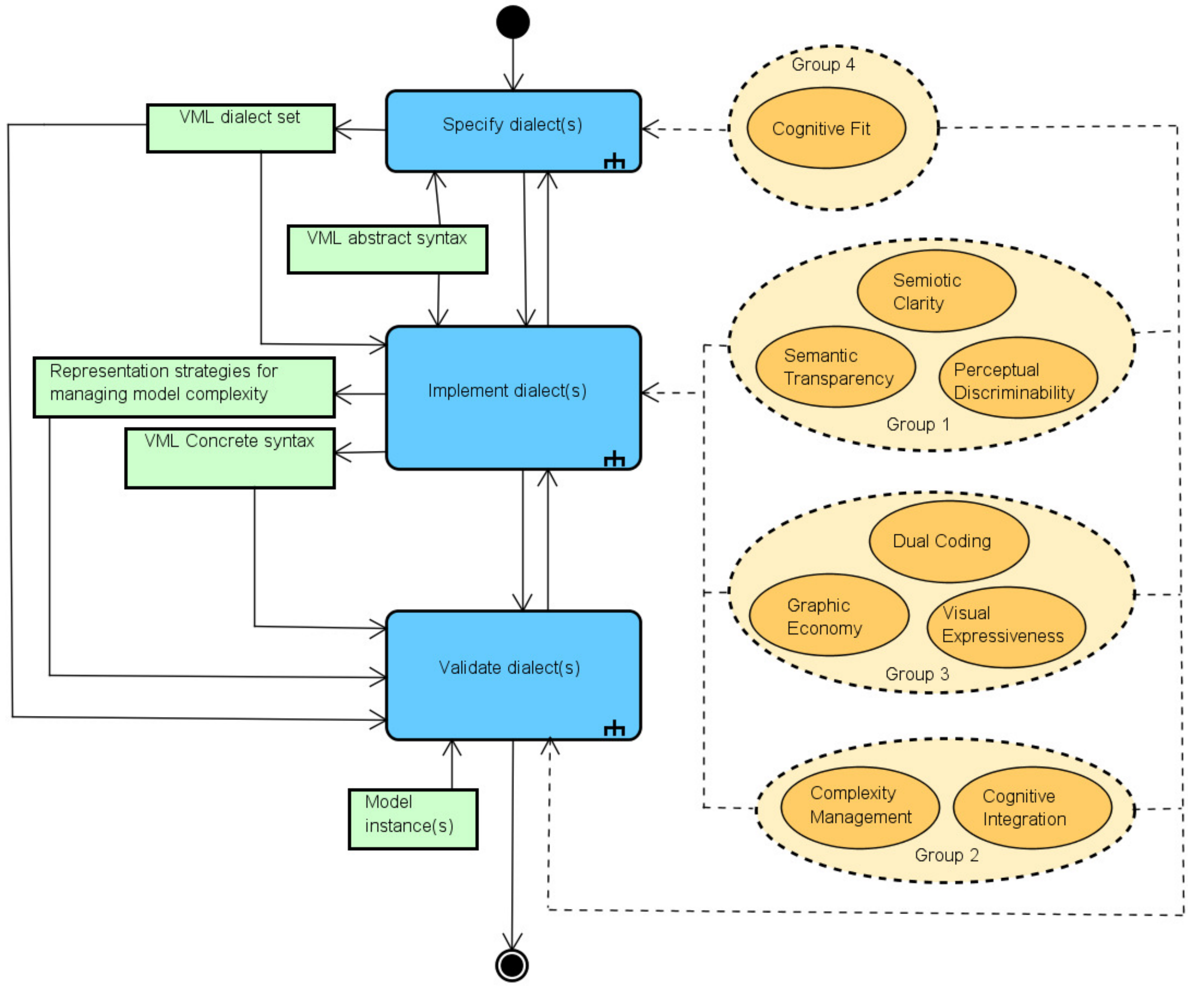


Figure 25: Design process overview

The *Specify Dialect(s)* phase is performed considering the language abstract syntax (as input) and the Cognitive Fit principle (as guideline), producing as output the VML dialect set. Identifying this information in the beginning of the design process aids the designer in assuming suitable commitments when developing the symbol set and representation strategies for the language.

In the next phase, *Implement dialect(s)* (see Figure 27), each dialect identified should have its set of symbols defined in accordance to the goal and directives previously identified. This phase starts by analysing the language abstract syntax and the language dialect set. This analysis results in a better understanding of the language metamodel and characteristics before designing the concrete syntax.

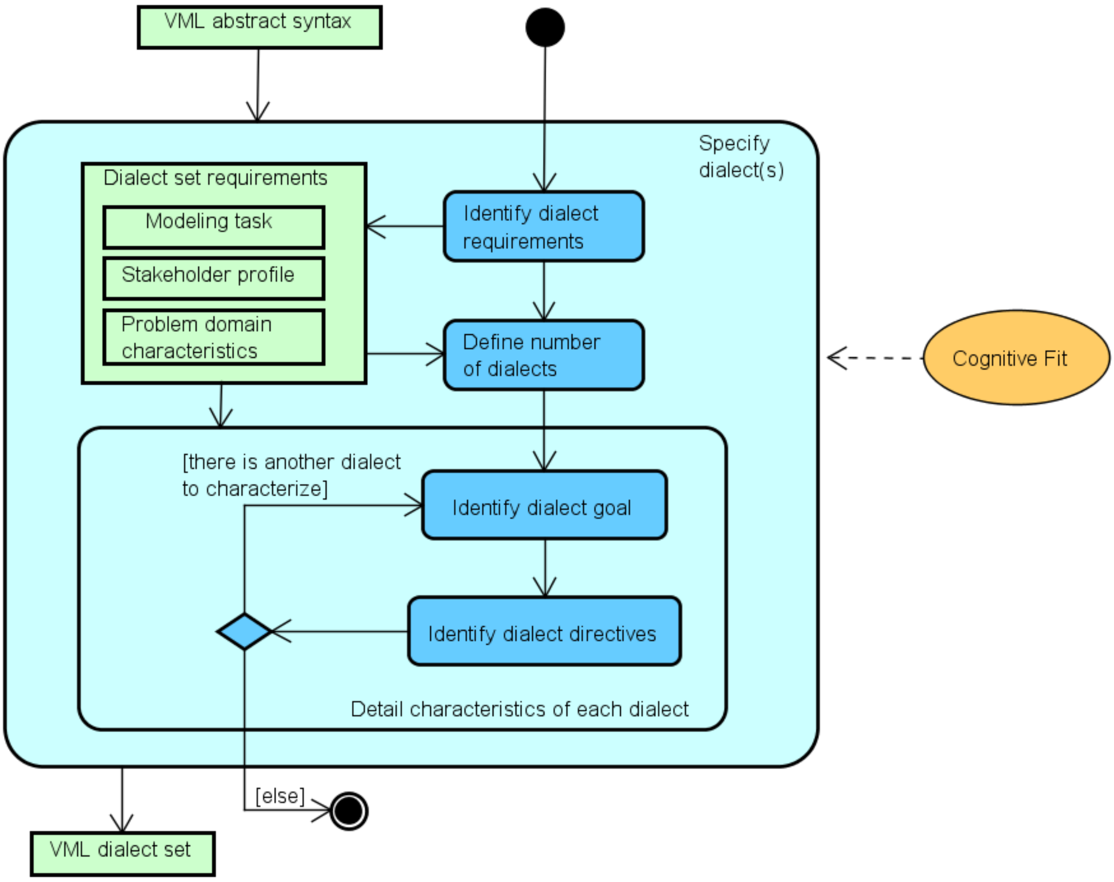


Figure 26: *Specify dialect(s)* phase

Next, the designer has to *Define dialect symbol set*, which is responsible for defining representational elements for the model elements. This is a compound activity, which is detailed in Figure 28. After defining a symbol set, optionally, support principles can be applied (see Figure 29). Finally, when the amount of elements requires managing model complexity, the *Identify ways to manage model complexity* activity (see Figure 30) shall be performed.

The input for the *Implement dialect(s)* phase are the language dialect set and the abstract syntax. The outputs are the VML concrete syntax and, optionally, some representation strategies to deal with size and complexity of the diagrams.

Figure 28 depicts the activity for defining the dialect symbol set for each dialect previously identified. This activity starts by choosing a model element to be represented (*Choose a model element to be represented* activity). This activity is guided by the Semiotic Clarity principle to ensure that each model element will be represented by exactly one symbol, unless this situation is required due to the directives established for that dialect. Once the model element to be represented is chosen, the designer needs to define a symbol for it (*Associate a symbol to the model element* activity). This activity is guided by the Semantic Transparency principle in order to establish a clear meaning for the symbol. Also, the designer should relate the chosen symbol to other symbols already defined in the concrete syntax, following the Perceptual Discriminability principle (*Relate new symbol to already defined symbols* activity). This activity aims at establishing the visual distance between the new symbol and other symbols already defined. These three activities are performed in a loop until all the representational elements of that dialect have been defined.

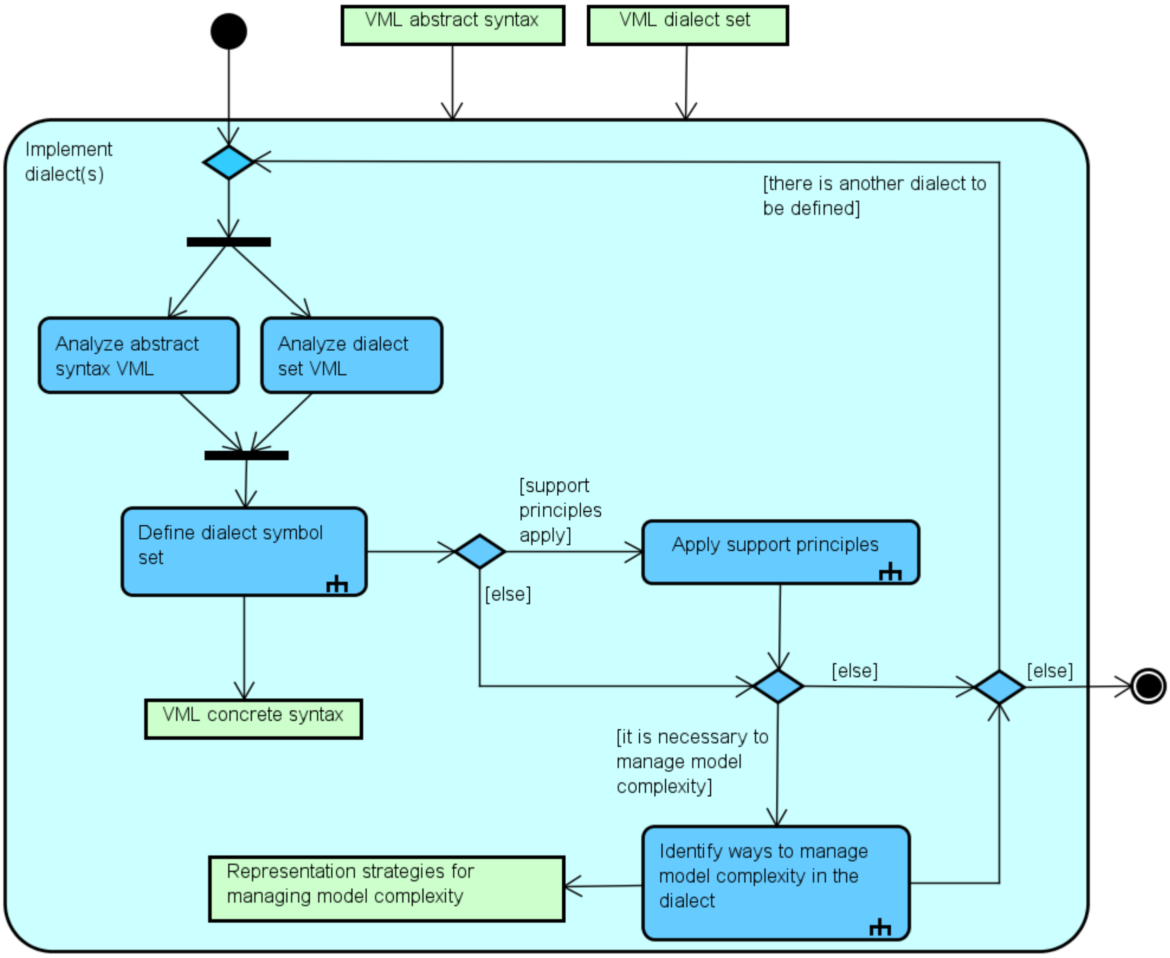


Figure 27: *Implement dialect(s)* phase

The *Apply support principles* activity (see Figure 29) deals with the possible application of the supporting principles: Visual Expressiveness, Graphic Economy and Dual Coding. The designer can apply each principle as much as s/he deems necessary. There is no pre-defined order to be followed. The inputs are the language abstract syntax, the concrete syntax and the characteristics of the dialects. The output can be an update of the language concrete syntax or of some representation strategy for managing model complexity.

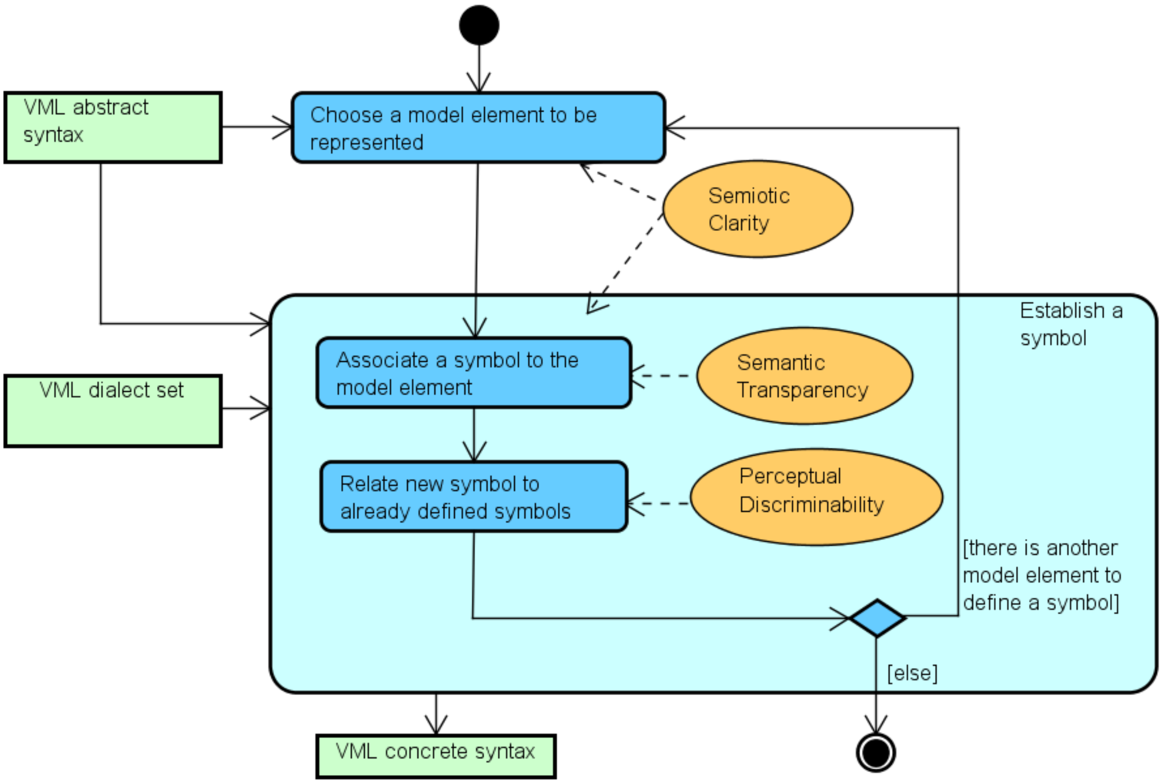


Figure 28: *Define dialect symbol set* activity

In the *Improve use of visual variables* activity, the designer shall review the symbols, possibly updating the visual variables values to maximize their expressiveness. The *Simplify symbol set activity* focuses on reviewing the symbols, now with the goal of simplifying the dialect. Finally, in the *Define textual complement* activity, the designer should evaluate when it is useful to introduce redundancy by using text. This can be necessary when the designer deems that text will increase symbol expressiveness.

After defining the symbols of a dialect, the designer should decide if it is necessary to manage the model complexity in diagrams developed using this dialect. Therefore, the *Identify ways to manage model complexity* activity is an optional activity whose importance increases as the language grows in size and complexity. Figure 30 details this activity, whose outputs are the language concrete syntax (in case it suffers some update) and representation strategies for managing model complexity (as many as the designer deems necessary).

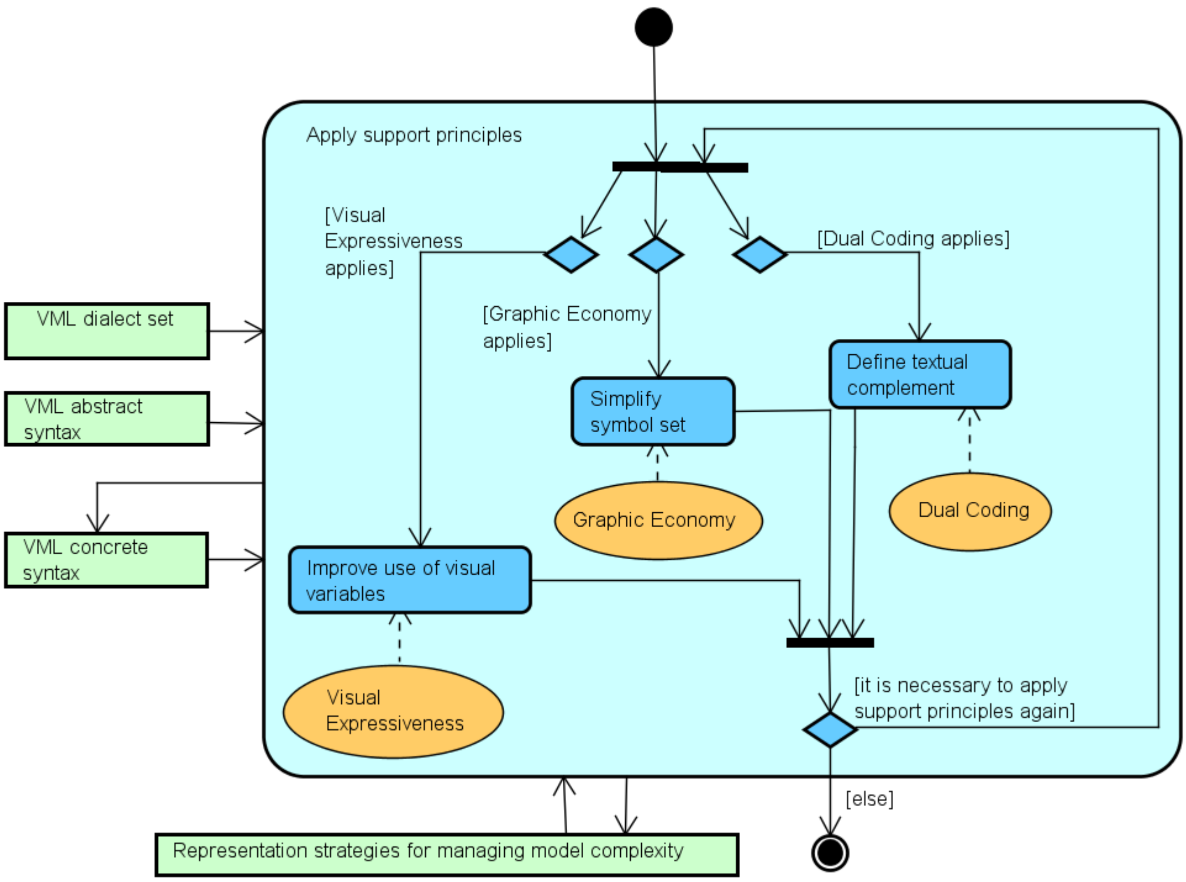


Figure 29: *Apply support principles* activity

First, the designer should *Evaluate complexity of symbol set* to deeply understand its characteristics[25]. Next, s/he shall *Manage model complexity*, which is an activity guided by the Complexity Management principle. In this activity, representation strategies for managing the complexity of diagrams written in that dialect are established. An example of strategy is the use of modularization. Also, the designer shall perform *Integrate information from different diagrams*, an activity guided by the Cognitive Integration principle. This activity is responsible for establishing ways to trace information spread in several diagrams and strategies for connecting them. These two last activities can be applied in parallel, resulting in a single representation strategy that is in accordance with both aspects of complexity management (organization and integration of information). In fact, both activities are applied in independent loops until deemed sufficient by the designer. Usually, each cycle results in a representation strategy for managing model complexity, which is

[25] There are different methods that can be applied in this activity, as for example the one suggested in (SCHALLES, 2013).

complemented by new representational elements, when necessary. As a manner to increase the quality of representation strategies identified at this moment, optionally, they can be subjected to the application of supporting principles (*Apply support principles* activity).

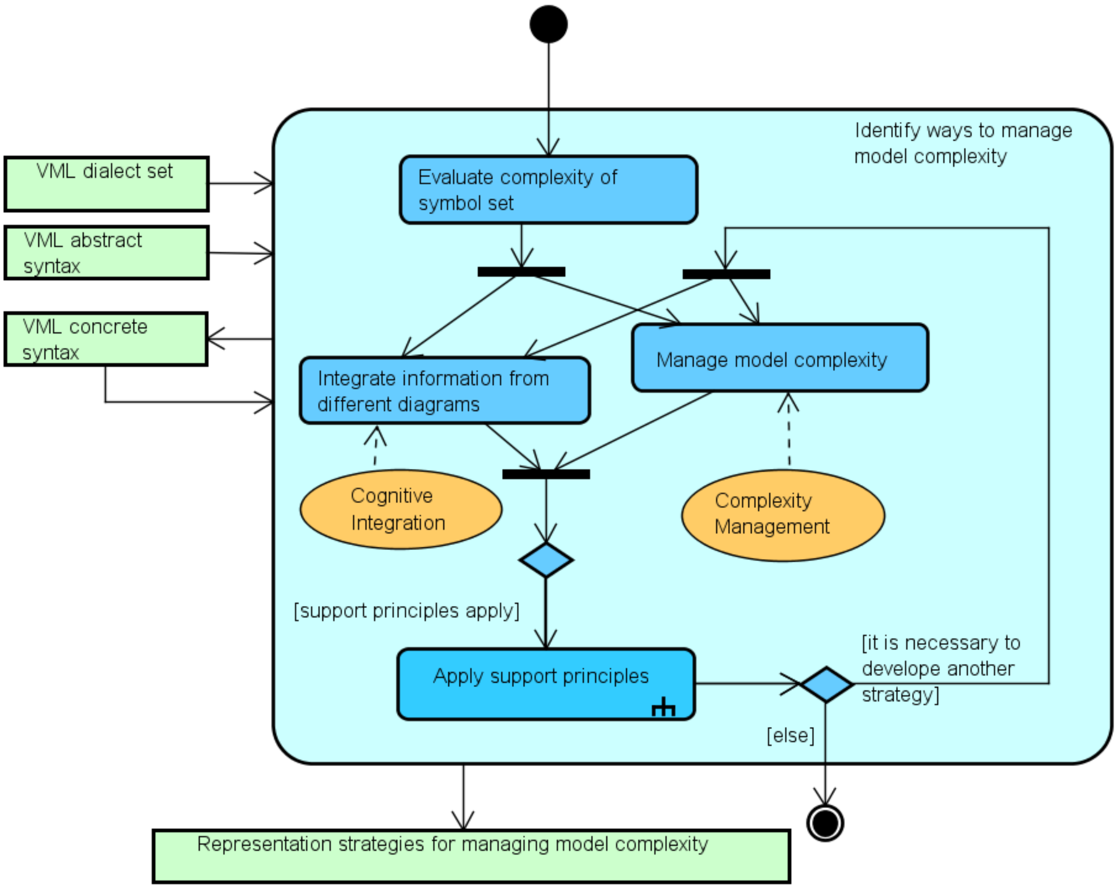


Figure 30: *Identify ways to manage model complexity* activity

Figure 31 details the *Validate dialect(s)* phase, which should be applied to each dialect. If problems are detected, the *Implement dialect(s)* phase should be performed again to solve them. If the problems are detected during the evaluation made by potential language users, a return to the *Specify dialect(s)* phase can be required.

The validation phase may involve two types of evaluation: one applied by the language designer, and another performed by means of empirical studies. The first one is a mandatory activity divided into two sub-activities: (i) *Check PoN principles' conformance*, which is responsible for assuring that PoN theory is respected in the proposed notation for a dialect, unless there is a different recommendation in the dialect characteristics; and (ii) *Evaluate diagram instances*[26]. This last activity demands another input: model instance(s), that is, models generated to have their concrete syntax evaluated through the instantiation of diagrams - as many as the designer deems necessary. At this point, the designer should generate diagram instances using the dialect and observe if the result conforms to the dialect goal and directives. The designer can select some quality properties to determine the quality of the resulting dialect, as the ones briefly commented in Section 2.1.

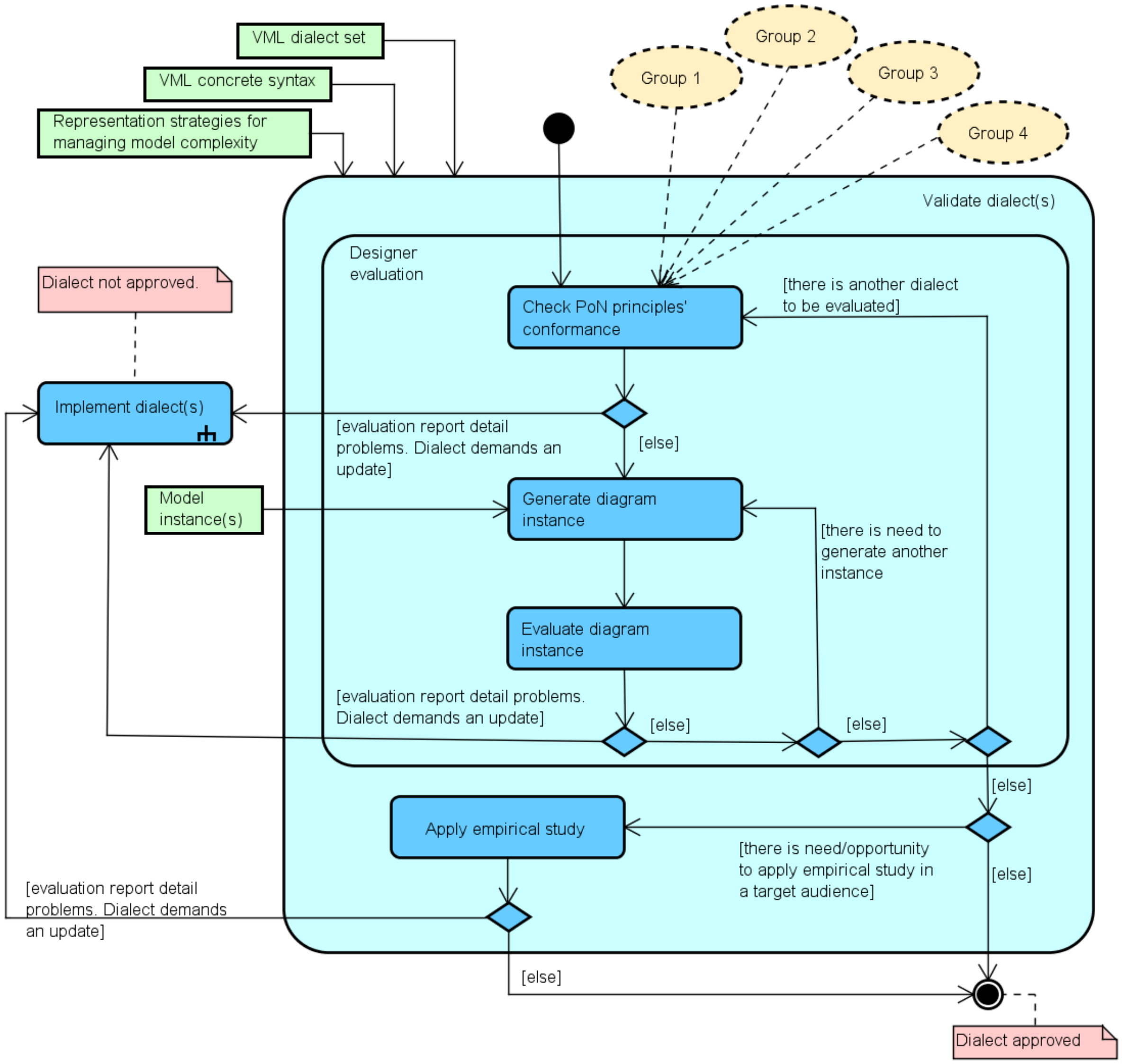


Figure 31: *Validate dialect(s)* phase

[26] A diagram instance is any diagram elaborated using the language. It is different of an instance elements diagram, a diagram which elements are instances of types elements (entities and relations).

The empirical evaluation performed with the target audience of the language, in turn, is an optional activity. This activity intends to analyze impressions of the target audience when applying the dialect. At the end of this group of activities, the designer should have evidence regarding the concrete syntax quality.

Depending on the problems detected in each of validation activities, the evaluation of the dialect can be assessed to be positive or negative. However, even in a positive evaluation, small problems can be detected. In case of a positive evaluation without identifying problems or if benefits of correcting the detected problems do not justify the effort to be undertaken, the dialect is considered approved; otherwise, the dialect is not approved and a new cycle of design process must be started.

4.4 Case Study: Fragment of the OPL Language, PoN-S release 1

This section presents a case study performed to evaluate the feasibility of the developed approach (PoN-S). Considering that the design task is a complex task, we planned to evaluate our proposal in an evolutionary manner, in each evaluation adding some characteristics besides the indications of the previous studies. This case study was our first evaluation. Besides, the evaluation of PoN-S, the case study also produced a new artifact: a DSML concrete syntax, an artifact of the designer interest.

The study aimed to answer the following question: Is PoN-S helpful in the design of modeling languages concrete syntaxes? This work is published in (SILVA TEIXEIRA *et al.*, 2016). The evidences collected here are considered in the current version of PoNTO-S (see Chapter 5).

Section 4.4.1 briefly introduces the modeling language under design. Section 4.4.2 describes execution of PoN-S release 1 in the design of the basis language concrete syntax.

We highlight that even if the DSML is in a certain sense an ontology-based language, the designer was not applying directly the ontological guidelines, even if she knows the baseline work (GUIZZARDI, 2013). This is the reason for us to consider this case study as an application of PoN-S instead of PoNTO-S.

4.4.1 OPL – a VML for the Domain of Ontology Development

An Ontology Pattern Language (OPL) is a network of interrelated domain-related ontology patterns that provides holistic support for solving ontology development problems for a specific domain. It contains a set of interrelated domain-related ontology patterns, plus a process providing explicit guidance on what problems can arise in that domain, informing the order in which these problems should be addressed, and suggesting one or more patterns to solve each specific problem (FALBO *et al.*, 2013)(FALBO *et al.*, 2016). For adequately representing OPLs, two types of models are necessary: a structural model, showing the patterns and the dependency relationships between them, and a process model, showing, among other things, the activities of applying the patterns, decision points, and entry and end points in the OPL process. The OPLs can favor the reuse of ontologies in activities of Ontological Engineering, speeding up the development process, as well as improving the quality of the resulting ontologies.

To facilitate the use of an OPL, the process and the relations between the patterns must be represented in a clear, unambiguous and complete manner. Visual notations can be used to provide a visual representation of the OPL, being an important means of communication between the stakeholders involved in the language use. This is the motivation for the application of PoN-S.

The visual notation for an OPL partially described here was proposed by Glaice Kelly Quirino in her master dissertation. Also, it is partially described in (FALBO *et al.*, 2016).

Regarding the process model, in the proposal presented here, its metamodel is an extension for representing OPLs of the metamodel of the UML activity diagram. For this reason, its concrete syntax is based on the UML notation for activity diagrams. This has the advantage of benefiting users who are familiar with this notation. However, in this section we do not discuss the design of the visual notation for the process model. Our focus here is on discussing the application of PoN-S in the design of the visual notation for the structural model.

Figure 32 shows the metamodel of the language concerning the OPL structural model. This model is composed of OPL Structural Elements. There are two types of OPL Structural Elements: Pattern and Pattern Group. A Pattern represents a domain-related ontology pattern, that is, a small and reusable fragment of an ontology conceptual model, extracted from a reference ontology (FALBO, R. D. A. *et al.*, 2013). A Pattern Group is a way of grouping related patterns and other pattern groups. Thus, a Pattern Group is composed by OPL Structural Elements. A special type of pattern group is the Variant Pattern Group, which is a set of (variant) patterns that solve the same problem, but each in a different way. Only one pattern from a Variant Pattern Group can be used in a given situation. Patterns that compose a Variant Pattern Group are variants of each other, giving rise to the derived relationship variantOf between Patterns.

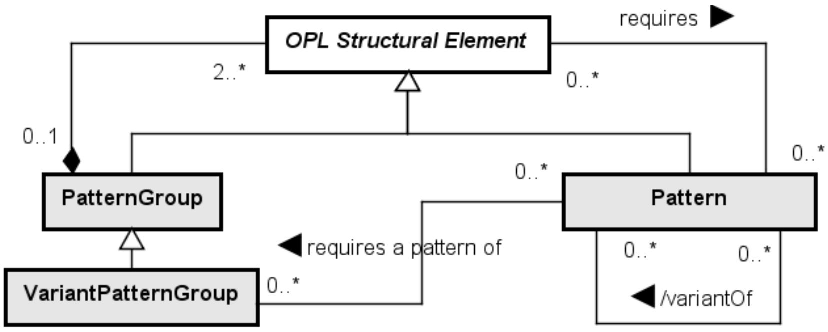


Figure 32: OPL structural metamodel

Patterns may depend on other patterns, i.e., for applying a pattern *p2* another pattern *p1* has to be applied first. An OPL should be able to represent dependencies between patterns or between a Pattern Group and a Pattern. The requires relationship captures this dependency. In the case of a dependency between a Pattern Group and a Pattern, the following rule applies: If a pattern *p1* is part of a pattern group *pg* and *pg* requires a pattern *p2*, then *p1* requires *p2*. Finally, a Pattern may require the application of a pattern from a Variant Pattern Group.

4.4.2 Execution

As Figure 26 and Figure 27[27] show, the design process started by identifying the dialect requirements, which include: (i) *Domain characteristics:* the visual notation for representing OPLs. Each OPL can refer to a different domain. Thus, this is the case of a domain-independent language; (ii) *Stakeholder profile*: OPLs are typically

[27] In PoN-S first release these two diagrams were reunited in one diagram. It was considered then the initial design process overview.

used by ontology engineers (both beginners and experienced); (iii) *Modeling task*: developing domain ontologies by reusing domain-related ontology patterns.

Although there are stakeholders with different levels of experience, the OPL visual notation should be simple and intuitive for all kinds of stakeholders. Thus, the designer established that only one dialect is enough. The goal of this dialect is to provide a simple and intuitive visual notation for ontology engineers to develop domain ontologies by reusing ontology patterns (FALBO *et al.*, 2013)(FALBO *et al.*, 2016). The notation should contain symbols to represent all OPL constructs without ambiguity. Moreover, in case of the use of colors, it should be possible to print the diagrams in gray scale without denting their comprehensibility.

The next step is to define the dialect symbol set. A loop was performed, in which each model element was characterized and had a symbol defined for it. This loop was guided by the principles of Semiotic Clarity, Semantic Transparency, Perceptual Discriminability as well as the supporting principles. Initially, considering the abstract syntax defined by the metamodel shown in Figure 32, and taking into account the Semiotic Clarity principle, a 1:1 correspondence between the metamodel constructs and graphical symbols was defined. This otherwise isomorphic mapping has two exceptions: the designer decided that it was not necessary to assign a symbol to the OPL Structural Element construct (an abstract modeling element), but only to its (concrete) subtypes (Pattern and Pattern Group). Moreover, symbols should be assigned to the relationships between these constructs, except for the variantOf relationship, since it is a derived association. Thus, symbols should only be assigned to the constructs shown in gray in Figure 32 and to the regular associations between them.

The designer started assigning a symbol to the Pattern construct. Since she was dealing with a domain-independent language, she decided to represent patterns by rectangles (an abstract sign). This choice was done considering that this is a common symbol used for representing patterns in Software Engineering Languages (for example, UML class diagrams). Concerning Semantic Transparency, on one hand, this symbol is considered semantically opaque, since it does not inform its meaning directly (MOODY, 2009). However, on the other hand, it can be considered a good design decision, given that this symbol is easily recalled (MOODY, 2009).

Pattern Groups are represented by figures closed by straight solid lines (solid polygons). For representing Variant Pattern Groups, the same notion was applied, but now using dashed lines. This decision was taken considering the Perceptual Discriminability principle, aiming at guaranteeing that symbols representing groups have a small visual distance. Furthermore, the visual variables *texture* and *color* were used to differentiate them, acting as redundant coding and increasing the discriminability between the symbols (but preserving their connection). The lines of Variant Pattern Groups are dashed and red, while the lines of Pattern Groups are solid and blue.

For representing the relation between Patterns and Pattern Groups, the designer chose the notion of spatial inclusion: Patterns that are part of a Pattern Group are represented as spatially enclosed by the symbol representing the latter. This choice affords the so-called *inferential free-rides* to the language, that is, visual querying and reasoning operations of minimal cognitive costs (GUIZZARDI, 2013). Moreover, it is noteworthy that there is a visual variable that qualifies Patterns and Pattern Groups: size. The region that represents the group encompasses several patterns. Thus, the size of this region is greater than the rectangle representing the pattern.

Regarding the dependency relations requires and requires a pattern of, both are represented by an arrow from the dependant to the dependee. For differentiating between them, arrows representing the requires association are symbolized with solid lines, in contrast to the dashed lines for the requires a pattern of association. This decision is in line with the one of representing Pattern Groups using solid lines, and Variant Pattern Groups using dashed lines. Thus, it takes the Perceptual Discriminability principle into account. So, these symbols have small visual distances.

It is worthwhile to point out that supporting principles were also applied for making the aforementioned choices. Regarding the Visual Expressiveness principle, the proposed visual notation uses the following visual variables: shape, texture and size. Color values are used as a redundant encoding, because variation in color disappears when a diagram is printed in grayscale. The designer decided not to apply other visual variables, keeping the notation as simple as possible.

The Graphic Economy principle did not play a strong role in this case study. This is because PoN advocates the use of up to six elements in a dialect and the structural metamodel considered here has only four classes and three regular associations. Nevertheless, some decisions were taken aiming at making the language as simple as possible. In summary, no symbol was assigned to the following metamodel elements: OPL Structural Element construct, since it is an abstract class in the metamodel (that is, it cannot be directly instantiated); whole-part relationship between Pattern Group and Pattern, since the notion of containment used to represent Pattern Groups also addresses this relation; and the derived association variant of, since it is also derived from the representation for Pattern and Pattern Group.

Finally, the Dual Coding principle, which deals with the use of text as an information supplement, was not applied. This is because, according to the designer: there is a small number of constructs to represent, their semantic is clear enough without textual redundancy and use of textual values can be better applied to distinguish between instances (as instance labels).

After defining an initial version of the concrete syntax, it is time to evaluate if the language demands representation strategies for managing model complexity. If this is the case, we should apply the principles of Complexity Management and Cognitive Integration. The Complexity Management principle emphasizes the importance of managing the diagrammatic complexity, which is measured by the number of elements in a diagram, among others. In the case of this case study, the designer recognized the need for managing complexity. Although the proposed language for representing OPLs is simple, the models that may be built using it tend to be large. Thus, to increase the speed and accuracy of understanding the diagrams, the designer decided to introduce a symbol for representing Pattern Groups (including Variant Pattern Groups) that encapsulates the Patterns that comprise it. Following the Perceptual Discriminability principle, the designer chose to represent these alternative forms by means of rectangles decorated by the following icon (⋔), indicating that this element is detailed in another diagram[28].

---

[28]This icon is commonly used by UML to represent that an element represented by the decorated construct encapsulates further elements. A similar symbol is used by the ARIS language (Software AG, [s.d.]).

Table 25 shows the final concrete syntax developed for representing OPL structural models.

Table 25: Symbols of the visual notation for OPL structural models

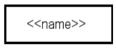
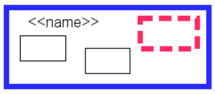
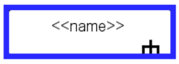
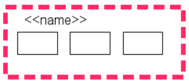
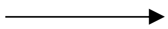
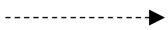
Structural Model	
Element	**Symbol**
Pattern	
Pattern Group (expanded format)	
Pattern Group (black box format)	
Variant Pattern Group (expanded format)	
Variant Pattern Group (black box format)	
Relation “requires”	
Relation “requires a pattern of”	

Figure 33 shows an example of a structural model of an OPL called Service OPL (S-OPL). This OPL, which provides ontology patterns for service modeling, is discussed in details in (FALBO *et al.*, 2016).

As shown in Figure 33, S-OPL is organized in three groups: Service Offering, Service Negotiation and Agreement and Service Delivery. The Service Offering Group is composed by three patterns (SOffering, SODescription and SOCommitments) and two groups of variant patterns (Provider Variant Group and Target Customer Variant Group). The patterns SODescription and SOCommitments as well as the Provider and Target Customer Variant Groups require the pattern SOffering. SOffering, in turn, requires patterns of both Provider and Target Customer Variant Groups. Provider and Target Customer Variant Groups are both composed of seven variant patterns each. The Service Negotiation and Agreement Group is composed by four patterns (SNegotiation, SADescription, HPCommitments and SCCommitments) and three groups of variant patterns (Agreement Variant Group, Hired Provider Variant Group and Service Customer Variant Group). The Agreement Variant Group is composed by three patterns: SNegAgree, SOfferAgree, and

SAgreement. The first two of these patterns as well as the SNegotiation pattern require SOffering. The patterns SADescription, HPCommitments and SCCommitments require a pattern of the Agreement Variant Group. The SAgreement pattern requires patterns of both Hired Provider and Service Customer Variant Groups (shown as black boxes in Figure 33). These two variant groups, in turn, require the SAgreement pattern. Finally, the Service Delivery Group (shown as a black box in Figure 33) requires the Service Negotiation and Agreement Group.

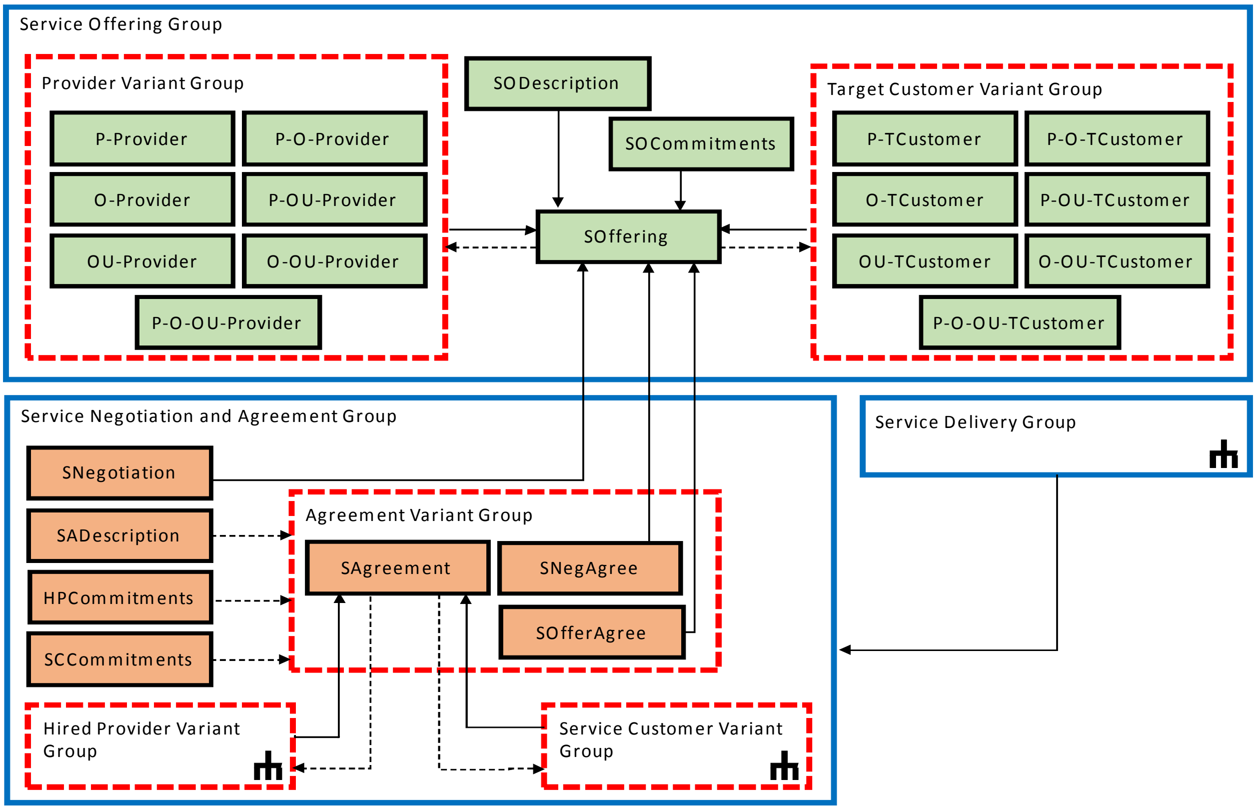


Figure 33: S-OPL structural model

4.4.3 Result

We asked the designer about her impression of PoN-S application. She indicated that PoN-S helped her in deciding how to design the concrete syntax. She would probably have done differently if she has no guidance. The resulting concrete syntax seems better in her opinion. This is in accordance to the main suggestion of the third exploratory study (see Section 3.3) – that a “step-by-step” could be helpful.

The designer applied a survey to test her proposal. As main result, she commented that the participants indicated that, in general, they had a low difficulty degree for understating how the OPL works, as well as, they understood the majority of the visual notation symbols. In addition, no problems were reported in the new representation of the OPL and the new representation was considered better than the previous one. However, this survey was just to collect first impressions on the proposed notation. Only an in-depth empirical study could verify if the OPL concrete syntax is good enough to be applied and probable updates in it.

It was during this study that arose for the first time the idea of dialect evaluation through building of diagram instances. Also, the designer suggested we should detail better the specification of requirements (the first phase of the design process). We adopted both suggestions in the next release of PoN-S.

4.5 Final Considerations

In a brief literature review, executed to identify how concrete syntax of conceptual modeling languages have been evaluated and designed, we identified PoN as the most widespread approach for analysis and design of VML concrete syntax (STORRLE; FISH, 2013),(GENON *et al.*, 2011). Also, we noticed that studies discussing efforts in analyzing modeling languages (with associated redesign suggestions)(for example, (GENON *et al.*, 2010),(FIGL; DERNTL, 2011)) are more common than those describing efforts in language design (for example, *(MISKE et al.*, 2014)).

The need for improving the design process involving PoN has been identified by many researchers, including Moody himself. In (MOODY *et al.*, 2010), Moody *et al.* discuss operational issues of PoN when presenting the analysis and redesign of i*. However, these issues are discussed individually for each principle, that is, the authors do not define a process involving all principles. In (CAIRE *et al.*, 2013), a work complementing the i* evaluation described in (MOODY *et al.*, 2010), the authors added the idea of PoN operationalization, highlighting the importance of considering stakeholder profiles during language design. It is a clear contribution towards considering pragmatic issues for notation analysis and design. However, once more, they did not define a design process.

In (STORRLE; FISH, 2013), Storrle and Fish criticize PoN judging that it still needs improvements towards operationalization. In that article, the authors propose ways for operationalizing PoN focusing on the analysis task of modeling languages.

Moreover, they established a series of measures that complement the PoN original proposal adding details to the principles that aid the analysis of concrete syntaxes. However, they also do not propose a design process.

The approach proposed here contributes to this collective effort of proposing *operationalizable* techniques for the design of visual languages. In particular, PoN-S is a methodological contribution that supports language designers in the application of PoN through the definition of a design process, a gap that has been identified in the literature.

However, it is not yet a finished work. We claim that some design activities can receive more details, providing a better guidance to a designer. A more detailed description of the activities is at disposal in the Appendix B, but even in this case we do not consider it enough. Some of the limitations are solved next, in Chapter 5, through the ontological guidelines. Other limitations demand future works (see Section 6.4). Also, we should invest in evaluation to prove or refute our decisions, as well as to improve the approach. This more in-depth evaluation is also considered a necessary future work.

We had two evaluative returns till this point: (i) The case study described in Section 4.4; (ii) The comments of the reviewers of the conference in which a paper summarizing this chapter was published ("EMMSAD'16," 2016). In the latter case, the main suggestion is the need to invest in evaluation as a manner to prove and to improve PoN-S.

Chapter 5. Physics of Notations Ontologized and Systematized (PoNTO-S)

After an initial case study (see Section 4.4) and discussions involving the researchers, we have confirmed that some of the aspects in PoN-S required further development. We advocate that the systematic guidance offered by PoN-S is necessary but not sufficient for a fuller methodological support in designing VMLs. This conclusion is in line with some limitations of PoN pointed out in Section 2.4: The establishment of the symbols, connecting them to the real-world semantics.

To overcome these limitations, PoN-S is extended to consider ontological guidelines, giving rise to the ***Physics of Notations Ontologized and Systematized (PoNTO-S[29])***. The main difference between PoNTO-S and PoN-S is the explicit consideration of ontological aspects in the former but not in the latter. Due to this in-depth guidance we decided to focus on DSML instead of VML (that contemplates DSML and DIML) in a first moment.

The addition of ontological guidelines in the design activity termed *Establish a symbol* solves the limitation of PoN which indicates that establishing a solid connection between the real-world concepts and the symbols that represent them is a good design decision, but PoN does not detail how to do this connection. We suggest doing this through ontological properties. Also, in this way we are solving another shortcoming: the insertion of the baseline of ontological guidelines defined by Guizzardi (GUIZZARDI, 2013) in a design process. Complementing these expansion, the ontological guidelines are expanded to include new OntoUML constructs and to be considered in other design activity (*Choose a model element to be represented*), in this manner, partially solving the other identified limitations – once ontological guidelines compose the design process, we expanded their application keeping their coherence.

[29] The word "pontos" in Portuguese means "dots" as in the dots that establish a connection between different elements. We propose PoNTO-S as a methodology to "connect the dots" relating PON-S (an Information Visualization theory) with aspects of real-world semantics (through ontological guidelines).

Moody (2009) admits an influence of ontological analysis (avoiding semantic anomalies) when mapping ontology concepts (the real-world domain) to language constructs, but this activity is focused on abstract syntax instead of concrete syntax. However, Moody claims that it is desirable to establish a similar manner of evaluating and designing the mapping between model elements and representational elements. Thus, by defining symbol anomalies, PoN indirectly considers ontological analysis in the design of concrete syntax, although this is grounded more in Semiotic than in Ontology.

There are at least two manners in which an ontology can influence the design of concrete syntaxes of VMLs:

- An influence caused by the use of ontological properties as basis to guide visual notation design decisions. As for example, Guizzardi proposed that variation in `phase` entities can be represented as variation in color values (GUIZZARDI, 2013). This is the most important influence;
- An influence caused by the reference ontology that gives rise to the abstract syntax, that is, how domain aspects influence design decisions. The connection between the model elements should be mapped to the representational elements. That is, the structure of concepts existing in the ontological model should be mapped to the correspondent symbols, replicating somehow the ontological structure. As for example, to map a taxonomy of semantic constructs to an equivalent taxonomy of symbols (see Figure 15). In this case the ontological model that underlies the abstract syntax is being respected (GUIZZARDI, 2007)(HENDERSON-SELLERS, 2011).

PoN conceptualizes three spaces in the process of designing concrete syntaxes – problem space, design space, solution space (MOODY *et al.*, 2010). PoNTO-S considers ontological influence in two of these spaces:

- Problem space: The abstract syntax should be based on a reference ontology developed using an Ontology-Driven Conceptual Modeling Language (for example, OntoUML), as suggested by (GUIZZARDI, 2013). This guarantees that the ontological commitments underlying the model are exposed by the abstract syntax of the modeling language;

- Solution space: The mapping between model elements and representational elements is influenced by the reference ontology and the adopted ontological theory (UFO), as suggested by (GUIZZARDI, 2013).

This chapter is structured as follows: PoNTO-S added ontological guidelines to the PoN-S design process. These guidelines are described in Section 5.1. Section 5.2 describes two empirical studies performed during the development of PoNTO-S.

5.1 Ontological Guidelines

The ontological guidelines considered by PoNTO-S are based on the ones described in (GUIZZARDI, 2013), but also include extensions and improvements. In PoNTO-S, Guizzardi recommendations are restructured in a guideline format and extended when necessary. Some extensions made in the original ontological guidelines are: (i) PoNTO-S includes an alternative representation for `roles`, when `relators` are represented; (ii) When mapping the `phase` construct, Guizzardi suggests applying color, brightness or texture. PoNTO-S establishes when it is suitable to apply each of these visual variables and how to deal with `phase` hierarchies; (iii) Guizzardi identifies that sometimes it is not necessary to represent abstract concepts. PoNTO-S provides guidelines for deciding when to represent, or not, abstract concepts; (iv) PoNTO-S also discusses whether representing `roles` or abstract concepts can be considered cases of symbol deficit. Additionally, some new guidelines are added, such as the ones that guide the establishment of a sequence for choosing model elements to be represented.

Currently, PoNTO-S considers guidelines based on a sub-set of OntoUML constructs, namely: (i) `kind`, `subkind`, `phase`, `role` (indirect representation through the representation of the `material` relation and the involved `sortals`), `material` relations, `componentOf` relation. These guidelines are based on the ones presented in (GUIZZARDI, 2013); (ii) `role` (direct representation of this construct), `relator`, and `mediation` relation, which are included in PoNTO-S to establish a core set of UFO concepts commonly used in developing OntoUML models[30]. We decided to define these new guidelines based on the indications of the explorative empirical studies and the work of Sales and Guizzardi (2015), who built an OntoUML benchmark including 54 OntoUML models. From this benchmark, it is possible to see that among the OntoUML constructs most used in the models are `roles`, `relators` and `mediation` relations (see Appendix C for more details).

The resulting ontological guidelines are integrated to PoN-S process, more specifically to support performing the *Define dialect symbol set* activity[31], a core activity in the design of the DSML concrete syntax. Two sets of guidelines are proposed, helping: (i) To choose a model element to be represented; and (ii) To establish a symbol for each of these model elements. These groups of guidelines can be adopted in the corresponding activities identified in the design process, as Figure 34 shows. This figure extends Figure 28 by adding another type of guideline (in addition to PoN principles and their groups): the ontological guideline.

5.1.1 Ontological Guideline: Choosing a Model Element to be Represented

When choosing a model element to be represented, two questions arise: (i) Is there an ideal order for choosing elements to be represented? (ii) Should abstract entities be represented?

When addressing the elements to be represented, the following guidelines should be considered, respecting the order in which they are presented (for more detail see Appendix B):

(a) Entities and relations: Relations, by their nature, involve entities. So, entities should be represented first;

(b) Rigid (`(sub)kind`) and anti-rigid (`phase, role`) types: Instances of rigid types will continue to be so as long as they exist in a model (GUIZZARDI, 2005a). On the other hand, anti-rigid types are contingent specializations of

---

[30] We also developed a draft of an ontological guideline do deal with `collective` construct and `memberOf` relation. However, as we did not evaluate them through empirical studies they are exposed in Appendix A.

[31] We suppose that ontological guidelines can be applied in other design activities than the *Define dialect symbol set*. We comment on this in Section 6.4.

rigid types (GUIZZARDI, 2005a). So, rigid types should be represented first, followed by anti-rigid types.

Two alternatives for the representation of the `roles` are considered. In the first alternative, for the sake of graphic economy, the suggested guidelines have an option to refrain from directly representing `roles`. In this case, `role` should only be represented via the representation of the relationship between the entities involved in the relation. Obviously, in this case, there is no need to identify an order for representing this construct. In the second alternative, instead, `roles` are explicitly represented. The justification for this second alternative is presented when discussing the representation of `relators`;

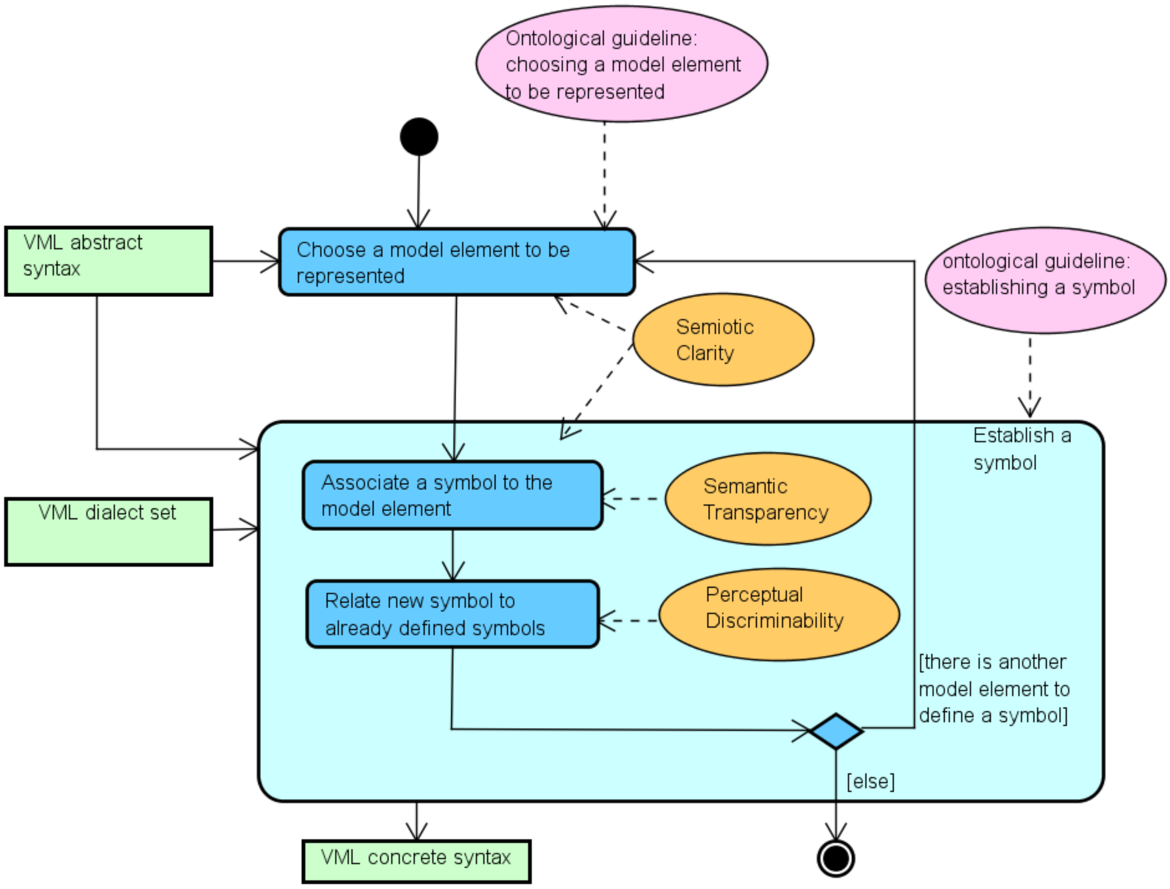


Figure 34: Define dialect symbol set activity - Inclusion of ontological guidelines

(c) Hierarchies: A specialized entity type adds details to the element it specializes. So, it makes sense to represent first elements in the top of a taxonomy, going down towards the bottom. For instance, in a kind/subKinds taxonomy, first the designer should represent kinds, and then its subKinds;

(d) Relator, roles and mediation relations: these constructs are strongly connected and bear dependence relations to each other (GUIZZARDI, 2005a). To make explicit this connection, they should be treated together. As roles are specialized from other sortal types, they should be represented after representing their supertypes. In addition to this, as roles can be involved in relations other than mediation relations, these roles should be represented before these relations, keeping in mind the idea that entities involved in a relation should be represented before the relation itself;

(e) Parthood relations (componentOf) and material relations: parthood relations refer to the structure of an entity, while material relations have a behavioral aspect relating entities, that is, these entities are involved in some event or action. Thus, it is suggested that parthood relations should be represented before material relations – preserving the intention that structural aspects should be represented before behavioral aspects (this idea follows the same rationale we make when suggesting that entities should be represented before relations). To make the design process more flexible, the designer may decide to represent parthood relations in a moment close to the design of the entities involved in these relations, considering that they have a strong connection and that the relation can interfere in the choice of the symbols to be used to represent these entities. There is no impact in the order in which the parthood relations are represented, that is, it is an unconstrained designer choice.

When selecting an entity type to be represented, the designer should consider if it is a concrete or an abstract type. If it is a concrete type, the designer should select a symbol to represent it. If it is an abstract type, the designer might decide to assign or not a symbol to it. This decision should take into consideration the language requirements, as for example the possibility of representing an abstract element in some diagram or even if the designer intends to represent specialized types of the abstract type through adding of details to the main symbol, making commonalities explicit in the abstract type representation.

The rationale supporting this guideline is that a `(sub)kind` can be an abstract type when it is, for example, in the upper levels of a complete generalization set of `subkinds`. In this case, it will never be directly instantiated. So, it is not necessary that this `(sub)kind` has a direct symbol to represent it. An abstract `(sub)kind` can be indirectly represented through the representational elements associated to its concrete `subkinds` (GUIZZARDI, 2013). In a similar way, a `phase` can be an abstract entity when involved in a complete generalization set of `phases`. In OntoUML, a `role` can also be an abstract entity when involved in a complete generalization set of `roles`. As previously discussed, in a first alternative for representing `roles`, it is suggested that these metatypes should not have a direct representation symbol. In the second alternative, given the need to explicitly represent `relators` and `roles`, `role` representations should be established. That is, it is not necessary to discuss at this moment whether abstract `roles` should be represented or not. This decision has to be already taken when choosing alternatives 1 or 2. If alternative 1 is selected, then no `role` (concrete or abstract) is represented. If alternative 2 is chosen then all `roles` (concrete and abstract) should be represented.

In the defense of representing abstract types, Souza *et al.* (SOUSA, VANDERDONCKT, HENDERSON-SELLERS & GONZALEZ-PEREZ, 2012) claim that abstract entities can be instantiated in specific situations to avoid modelers making premature commitments. This is valid when diagrams are successively detailed till the necessary degree of detail is achieved. Thus, initial diagrams can have entities that are not totally defined yet. In these cases, abstract entities can be temporarily represented till its concrete subtypes are introduced in the model.

It is important to say that the guidelines described here are influenced by the Semiotic Clarity principle. According to this principle, a designer should avoid, or at least reduce, symbol anomalies – excess and deficit - when selecting a model element to be represented (MOODY, 2009). In PoNTO-S, symbol excess is not possible, since only elements in the metamodel can be chosen to assign a symbol.

Concerning symbol deficit, however, depending on the goals and directives established to the dialect, it can be accepted, although, in general, it is not recommended. In particular, PoNTO-S does not consider as symbol deficit a situation in which abstract types are not directly represented, but only via the representation of their concrete subtypes. This anomaly is considered to occur only when a modeling element that can be instantiated does not have a symbol associated to it. Furthermore, in one of the alternatives for representing `roles` (alternative 1), the authors deliberately allow for having them implicitly visualized through the `material` relation in which they are involved. This is a potential symbol deficit situation that PoNTO-S admits. This is done in benefit of the Graphic Economy principle.

5.1.2 Ontological Guideline: Establishing a Symbol

As previously discussed, PoNTO-S considers ontological guidelines related to a partial set of OntoUML constructs: the ones based on Guizzardi work, previously described, and the ones introduced here. In the following, the last guidelines are presented.

The established guidelines usually describe representational elements for type elements. So, it could be supposed that they are suggested for type elements diagram construction only[32]. However, the performed empirical studies that evaluated the guidelines worked with instance elements diagram construction without problems. As we can observe in the results of the empirical studies, we claim that the evolution of the representation of type elements to instance elements representation is natural and similar. However, this issue should be investigated in an in-depth manner.

Relator, Role and Mediation Relation

`Roles` and `relators` are important and heavily used constructs for modeling relations in OntoUML. The usefulness of these constructs was object of study in an empirical study described in Section 3.2. In this experiment, the authors collected evidences that a complete representation of a relationship (including `relator`,

[32] The only guideline that refers directly to instance elements is the new guideline described in the current section.

`roles`, `material` and `mediation` relations) leads to a less error prone interpretation of conceptual models. However, the same experiment indicates that depending on the demands of an interpretation task, it is not necessary to use a complete representation for a relation, which can unnecessarily overload the diagram. So, there are various ways to represent a relation (WAND *et al.*, 1999), and a language designer has to decide which one to adopt in the language being designed. This decision belongs to the metamodel level, when the designer should decide which should be the modeling elements and their relations. However, it should be supported in the concrete syntax level too.

Taking this situation into consideration, we decided to offer flexibility of representation of relations in the ontological guidelines – the designer may choose among different options of representation of relations.

As identified in the empirical study, there are at least 4 ways of representing `material` relationships in OntoUML. The guidelines of the current release of PoNTO-S cover two alternatives[33]:

- To represent only the `material` relation directly and the `roles` indirectly, as originally described in (GUIZZARDI, 2013). This is a simplified representation of a relation;
- To use a more complete representation including `relator`, `roles` and `mediation` relations, but not representing the `material` relation.

The first alternative was selected because it was already adopted in the baseline of the PoNTO-S ontological guidelines. However, it is worth mentioning that a representation without `roles` may be not advisable, because it is less specific and more error-prone, although it is the simplest visualization. On the other hand, the authors do not deem necessary the complete representation (`relator`, `roles`, `material` and `mediation` relations) because it can cause visual clutter, being useful only in some specific situations.

---

[33] We assume that PoNTO-S first release should be stable before we cover all possibilities – tasks, ontological guidelines – That is why we reduced the ontological guidelines possibilities, including dealing with `material` relations complete alternatives.

The relator is the truthmaker behind a material relation (GUARINO; GUIZZARDI, 2015). In this sense, relators are ontologically prior to the relations derived from them. For this reason, by explicitly representing the relators, we can justify in certain situations not explicitly representing the material relation. In the empirical study described in Section 3.2 we analyzed this type of variation in representing relations, among other possibilities.

Typically, relations are represented by lines with some property or change in them, while relators are rigid sortal types of existentially dependent entities. In other words, existential dependence aside, relator kinds are similar to kinds (that is, ultimate rigid sortals) like object kinds (GUARINO; GUIZZARDI, 2015). As kinds are considered as a central concept for representing entities, relators are central concepts for representing relations. Therefore, the suggestion made here is to represent relator kinds in analogous manner to which object kinds are represented, namely, by focusing on the shape visual variable. However, it might not be as easy as for the case of object kinds to find icons that represent with high perceptual immediacy relator kinds. Usually, relations are more abstract concepts than entities and therefore identifying them as symbols that represent a real-world situation is often a difficult activity. In summary, as a complement of the claim made by (WAND *et al.*, 1999) that relations are problematic to model, we extend this claim to: relationships are difficult to model and to represent.

Given these previous consideration, we suggest representing relators through abstract symbols (geometric figures), applying the Dual Coding principle (MOODY, 2009) to assist in identifying each relator kind. In addition, the choice of geometric shapes is purposeful to facilitate the combination with other symbols, which can be used to represent the roles (a geometric form is usually more abstract than an icon, making the concatenation easier with other symbols, as suggested for role representation - see text below). The suggestion allows that other types of symbols are used to represent relators, as long as the symbol has a closed form that supports combined representation of roles. For example, the index of a wedding ring can be used to represent the <u>marriage</u> relator.

Moody (2009) suggests using redundant coding to highlight or identify more clearly some symbols. Concerning this, for representing relators, the application of the color visual variable can be a good design choice. However, to help in differentiating between the representation of `phases` and redundancy in `relators`, it is suggested to apply color in the interior area of the `phase` symbol (it should have a black border and some color inside the contour), while for `relators` color should be applied in the border (while its inside color is left blank). This is in accordance to the Perceptual Discriminability principle.

The representation of `roles` includes three related aspects:

(i) The connection between the `role` and the `relator` that justifies its existence (representation of the `mediation` relation): `roles` are relationally dependent and, thus, they only occur in the context of `mediation` relations. Thus, attaching the shape of the `substance sortal` that gives rise to `role` to the `relator` representation is a manner of reinforcing this dependence.

Ware (2013) presents a visual grammar of relation representations, including whole-part relations. In this grammar, the author suggests using the partitioned region, attached shapes and enclosed shapes as manners to represent whole-part relations. PoNTO-S applied partitioned region and enclosed shapes for `componentOf` relations. However, it is not seen as a confound factor the application of attached shapes for `mediation` relation instead of a whole-part relation, considering that the attachment representation also can be used to reinforce a strong connection between the basic element and the attached ones – it is a somehow a whole-part relation;

(ii) Representing the `role` itself: merely placing the representation of the `substance sortal` that gives rise to the `role` as an element attached to the `relator` representation could be enough if that `substance sortal` is involved in a single relationship. However, usually this is not the case. In order to maintain the readability of the language, the symbol associated to the `substance sortal` should be duplicated in a reduced size and this reduced symbol should be attached to the `relator` representation. Furthermore, to strengthen the application of the Semantic Transparency principle regarding the connection between `relator` and `role`, the same aspect (color, brightness, texture) applied to the `relator` should be applied to the `role` (independently of the original shape of the substance sortal representation that originates the role).

Here a possible problem can be noticed: when roles connected through a relator are based on the same substance sortal, how can the modeler make clear the different roles? An example is given in Figure 35. In Figure 35(a) we have a simplified metamodel. In Figure 35(b), as instance element diagram, we can notice how it would be difficult to identify which roles Paul and Peter are performing. To avoid this problem, PoNTO-S suggests using the Dual Coding principle (identifying each role), and thus to show the names of the roles together with the miniature representing the role. As visual alternatives: (i) Adding same shape detail in the representation of the roles to differentiate between them; (ii) Appealing to the use of the texture visual variable, thus, keeping graphical differentiation between the two manifestations of roles. In Figure 35(c) we show how we can differentiate between the roles representation through the addition of shape details to identify clearly each role Peter and Paul are performing;

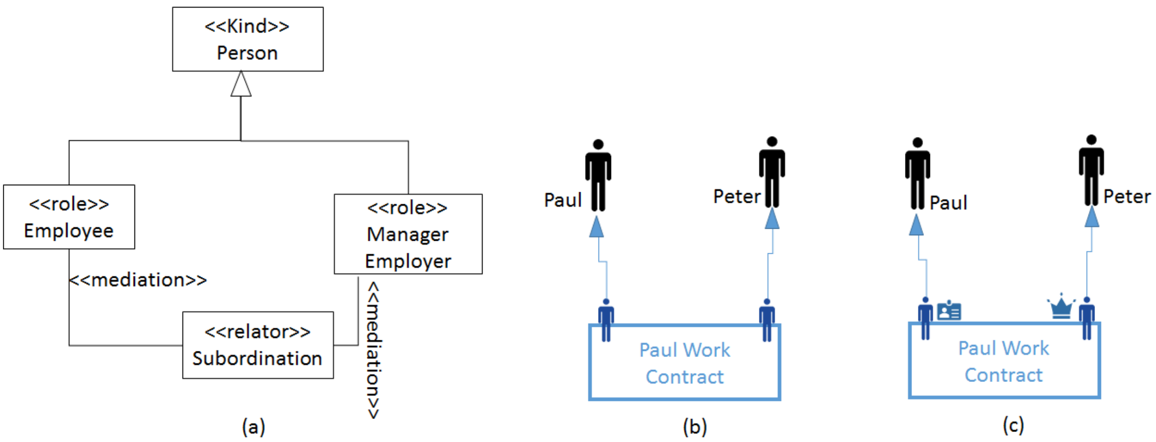


Figure 35: Example of role representation when the different roles connected to a relator are based on the same sortal

(iii) Representing the connection between the role and the entity type providing the instances that play that role, that is, the so-called allowed type (GUIZZARDI, 2005a): as this is a subtyping relation, the suggestion is to apply a similar representation of the UML representation of subtyping (a line connecting the elements and a small triangle in the association end of the line, near the representation of the allowed type). To increase the visual connection among the involved elements, the same color applied to the `relator` symbol and replied in the `role` symbol should be applied to the subtyping relation representation.

The suggestion above tries to diminish the quantity of different symbols used, reinforcing the Graphic Economy principle. On the other hand, it seeks to reinforce the Semantic Transparency and the Perceptual Discriminability principles.

Figure 36 presents an example of the suggested representation. Figure 36(a) depicts the metamodel while in Figure 36(b) the proposed visualization can be observed on an instance element diagram.

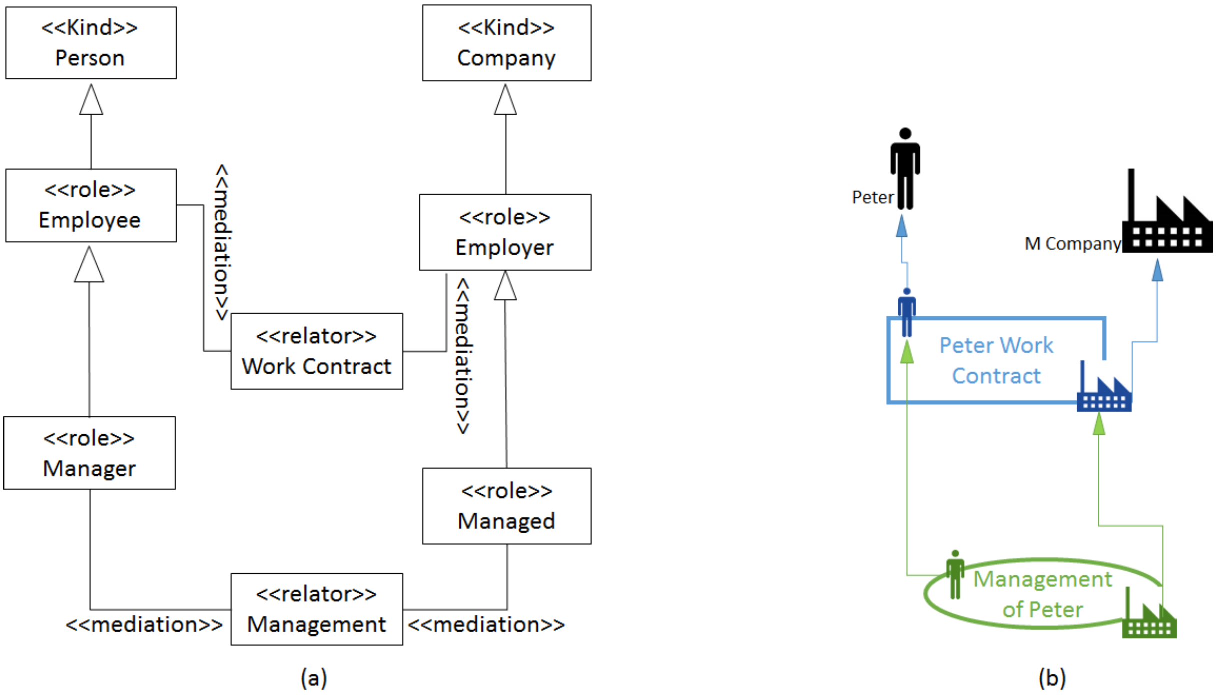


Figure 36: Example of a proposed concrete syntax for `relator`, `role` and `mediation` relation

We highlight that the above representation was proposed for type elements representation. However, the proposed representation is also applicable for instance elements representation, as we can notice in Figure 35(b) and (c). In this case what is actually visualized are `qua-individuals` (which in the case of `roles` are the characteristics that the `sortals` assume while performing that `roles`) - that is, in Figure 35 what we actually see is not a reduced symbol to represent Paul as an Employee, but the PaulQuaEmployee instance.

5.2 Evaluation

This section presents some empirical studies performed to evaluate the feasibility of PoNTO-S approach. Considering that the design task is a complex task, we planned to evaluate our proposal in an evolutionary manner, in each evaluation adding some characteristics besides the indications of the exploratory studies.

The studies aimed answering questions such as:

(i) Is an approach putting together PoN-S (a design process) and ontological guidelines helpful for the design of modeling languages concrete syntax? This issue is explored in Section 5.2.1.

The first object of study is the design process together with the original guidelines proposal in (GUIZZARDI, 2013). The purpose is to be sure that the design process and ontological guidelines can work together succesfully. The evidences collected here are considered in the current versions of both PoN-S and PoNTO-S.

(ii) Is PoNTO-S helpful in the design of modeling languages concrete syntax? This question is discussed in Section 5.2.2.

The second object of study is an improved PoNTO-S approach, resulting from analysis of previous empirical studies (as the case study of Section 5.2.1) and adding new ontological guidelines. A possible update of PoNTO-S approach due to some evidences collected in this study is identified as future perspectives (see Section 6.4).

5.2.1 Empirical Study: pre-release of PoNTO-S

This section describes an empirical study performed by a small group of participants applying a pre-release of PoNTO-S (basically, using PoN-S design process and the ontological guidelines proposed by Guizzardi (2013)). Its results allowed us to improve PoNTO-S (both the design process and the ontological guidelines), giving rise to the release described in 3.4s 4 and Chapter 5.

5.2.1.1 Objectives

The *research goal* was to analyze the design of a DSML concrete syntax with the purpose of characterizing the influence of PoNTO-S in respect to usefulness and usability of the approach in the viewpoint of novice language designers in the context of an organizational domain (basic concepts as Employee, Department, Commissions, superior-subordinate relation).

5.2.1.2 Design

The study was conducted based on the recommendations provided in (WOHLIN *et al.*, 2012).

The *research approach* was analytical, to collect early indications for further experiments and to obtain a first impression of PoNTO-S application. It was a qualitative experiment, as the researchers examined not only the representations proposed by the participants, but also their reasoning when applying the approach and their impressions on it. To collect evidence, the researchers analyzed the filled in forms (profile, feedback), the visual notation proposals, and applied a "think-aloud" protocol.

The *subjects* were three PhD students[34] of Computer Science field possessing knowledge of UFO. Two of them indicated not knowing PoN and one participant indicated that s/he knew PoN but never applied it. Two participants have similar profiles, they can be considered as experienced language users - model readers and writers. Both participants have more than 3 years of experience in UML and they know other modeling languages. One participant has a profile close to a model reader. S/he did not even know UML. None of them designed a concrete syntax before, so they can be classified as novice designers.

The task was the development of a concrete syntax applying PoNTO-S pre-release, in which the design process was an evolution of the process presented in (SILVA TEIXEIRA *et al.*, 2016) and the ontological guidelines of (GUIZZARDI, 2013) were formally considered. The participants had access to written instructions describing the process they should follow to create the concrete syntax. Inputs for the

[34] One of the participants are planning to do her enrollment on a PhD program, but s/he had not formalized the registration until the moment of the application of the experiment.

participants included: a fragment of the language metamodel written in OntoUML (the same one used in (SILVA TEIXEIRA *et al.*, 2013)) replicated in Figure 16[35], language goal and directives, and a model instance that could be used in the evaluation activity.

5.2.1.3 Results

Figure 37 and Figure 38[36] contain the concrete syntaxes proposals elaborated by the participants and a comparison proposal developed by the responsible researchers, generated in another experiment (see (SILVA TEIXEIRA *et al.*, 2013)). Some impressions collected from the table are:

- Even if there are similarities in the proposed symbols, there are no completely identical concrete syntaxes. This is the case even for concepts that are supposed to be universal (Person, Man, Woman). Participant 1 chose an abstract symbol to represent Person and applied details to differentiate between Man and Woman. Participant 2 applied indexes to represent only the `subkinds` (Man and Woman) – s/he associated a symbol for Person, but when drawing the diagram instance s/he changed her/his mind and decided that this symbol was not necessary. Both possibilities are in accordance to PoNTO-S recommendations. Participant 3 deduced that according to the domain there was no need to represent Person, Man and Woman, only the `roles` a Person perform in the organization. This was her/his justification to represent Employee as a stick man (an index usually associated to Person);
- `Phase` mapping is the guideline that is most uniformly applied;
- 2 out of the 3 participants did not represent relations. They were focused on entities;

[35] As the applied ontology in the current study is the same ontology applied previously in another empirical study (see Section 3.3), we checked if the current participants were not involved in the previous study or even if they knew the cited study.

[36] We divided the figure into two figures in order to increase the legibility of each proposed symbol.

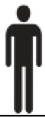
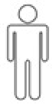
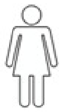
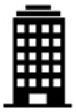
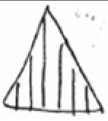
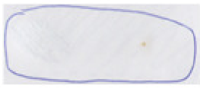
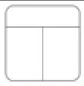
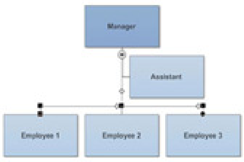
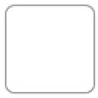
	Mapping: Model Element to Representational Element			
Concept	N1	N2	N3	Comparison Proposal
Person (kind)				No direct representation (abstract)
Man (subkind)				
Woman (subkind)				
Employee (role)				No direct representation (deduced by a relation´s representation)
Organizational Unit (kind)				No direct representation (abstract)
Department (subkind)				 Same sign of wholePart 2
Organizational Branch (subkind)			 Root of the tree (hierarchy)	
Commission (Kind)				No direct representation (abstract)

Figure 37: Concrete syntax proposal and comparison proposal. Part 1

- Concerning relations: (i) Participant 3 (the only one to represent relations) decided to apply hierarchical chart instead of spatial inclusion for representing the two whole-part relations. S/he argued that it was more natural to her/him. However, s/he changed her/his mind and decided to represent the relation between Commission and Commission member as indicated in the recommendation of PoNTO-S. When s/he was questioned about this issue – why representing the two whole-part relations differently - s/he indicated that s/he took this decision during the evaluation activity, when visualizing the resulting diagram;

- Participants tended to represent `roles`. When asked about this, they replied that even aware of the ontological guideline (the representation of `roles` can be indirect, throught the representation of the involved `sortals` and the `material` relation), they felt it was more natural to represent them.

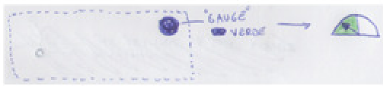
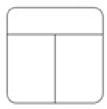
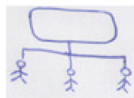
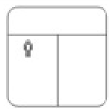
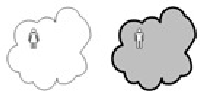
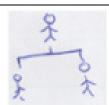
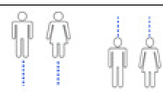
Normal Workload Commission (Phase)	 Color Blue	 Color Green	 Green gauge	
Overloaded Work Commision (Phase)	 Color Red	 Color Red	 Red gauge	
Superior Employee (Role)				No direct representation (deduced by a relation´s representation)
Subordinate Employee (Role)				No direct representation (deduced by a relation´s representation)
Commision Member (Role)				No direct representation (deduced by a relation´s representation)
WholePart 2 (Organizational Branch and Department)			 Convention, organization chart	
WholePart 1 (Department and Employee)				
WholePart 3 (Commision and Commission Member)				
reportsTo relation				

Figure 38: Concrete syntax proposal and comparison proposal. Part 2

5.2.1.4 Discussion

After analyzing participants proposals and studying the records generated by the think-aloud protocol, the researchers noticed:

- The ontological guidelines were the items more consulted during the activity execution, compared to other instructions at disposal, in this case the description of the design activities;

- We notice how the evaluation activities influenced the participants proposals. Sometimes, these activities conduced the participants to a different decision other than the initial;
- At several moments, the participants analyzed the given requirements. So, it is necessary to include this type of activity in the design process;
- In general, designers need to represent the `role` construct. So, in the next release of ontological guidelines it could be interesting to include this possibility;
- Representation of relations seems to be a problem. Two participants did not indicate any representation for any relation. They just needed them when drawing the diagram instance, choosing the traditional lines and arrows to do so (probably because this is the representation in the presented model instance). So, it is necessary to establish ways to improve the representation of relations. We hold the hypothesis that the representation of the `relator` construct can help in this sense, and it can allow a better representation of `role` construct;
- The application of the approach seemed to the participants harder than what we had initially assumed. We should identify a way to reduce effort in the application of the approach. Also, we should identify in which manners the approach could be made more flexible, considering different designer profiles;
- Designers have different profiles, as for example, some designer tend to be more focused on domains characteristics than in technical ones. This was the case of one of the participants in this experiment. Also, some participants prefer to follow the tradition in modeling[37], by representing boxes and arrows, and adding decorative icons to differentiate between entities. It could be interesting if the design approach could adapt to different designers' profile;
- Considering the sequence in which the participants chose what elements to represent in each moment, the most common way was by following the taxonomy of `kinds` – choose a `kind`, represent it, then choose an element in the next level of this `kind` taxonomy for representation. When a `kind` taxonomy is completely represented, go to the next `kind`. So, a guideline that

[37] It is probably easier for the participants to deal with a notation that they are used to.

deals with a sequence for selecting modeling elements for representation should take this possibility into account;

- It was a small sample size (only three participants). Although we could observe some indications, as the ones commented above, the experiment established restricted limits, as one dialect and a small fragment of a common domain. A more complete experiment is necessary to collect deeper evidence.

These issues were considered and influenced the generation of PoNTO-S release 1.

5.2.1.5 Limitations and Validity Threats

- Listed threat: We had three participants, so, it was a small sample size.

  Analysis after the empirical study execution: We considered that even with a small sample we achieved our purpose - to collect early indications for further experiments and to obtain a first impression of PoNTO-S application;

- Listed threat: We used as object of study a small fragment of a common domain.

  Analysis after the empirical study execution: As cited in the previous item, we considered that the purpose of the experiment was achieved. However, we are conscious that a more complete experiment is necessary to collect in-depth evidence;

- Listed threat: The think-aloud protocol is a good tool to record participants' behavior and impressions during the task, making the posterior analysis more complete. However, the participants may not demonstrate the behavior they would normally adopt when performing the task without such a direct observation.

  Analysis after the empirical study execution: We assured the participants that the recording would not be disclosed. Also, we only recorded audio instead of video. During the task, the researcher interacted with the participant as little as possible – only to remind them that they should express their reasoning aloud.

Also, each participant did the task separately. So, we believe that these measures reduced the impact of the threat.

5.2.2 Case Study: Fragment of LawV Language, PoNTO-S release 1

> “Law can be made more comprehensible if it is made more visual.” (OPEN LAW LAB)

The notion that visual communication can assist in the exercise of the Law profession is gaining form and we are able to identify publications (at different levels of formality) approaching this topic, which is being called ***Visual Law***. For example: (i) In (BRUNSCHWIG, 2014), Brunschwig describes trends in such a field; (ii) A blog discusses the question "What Lawyers need to learn from Information Designers" (HAAPIO; PASSERA, 2013); (iii) Another blog discusses how the use of illustrations can make complicated texts clearer (OPEN LAW LAB).

The goal of this case study was to generate a DSML for the Law domain, basing its abstract syntax on a domain ontology created as an extension of a core ontology (termed ***UFO-L***) (see Section 5.2.2.1), and designing its concrete syntax through application of PoNTO-S (see Section 0). This is a simplification of a complete DSML Engineering, as the ontology structure was considered as a complete and sufficient mapping to generate the abstract syntax (no rule was added and no update was studied). Also, only a partial evaluation of the DSML was conducted here (performed by the language designer). Even so, some interesting evidences were collected and these evidences can give rise to an improved DSML, called ***Law Visual Modeling Language (LawV)***, and to a new release of PoNTO-S.

The case study was performed by the language designer, Cristine Griffo, a PhD student who is developing the UFO-L core ontology as part of her doctoral research.

5.2.2.1 Description of the LawV Abstract Syntax

To develop the domain ontology that is the basis of LawV, the designer used UFO-L. The UFO-L[38] layer is a core ontology based on the concepts, properties and relations existing in UFO (GUIZZARDI, 2005a) and the rights structure of Robert Alexy Theory of Fundamental Rights (ALEXY, 2009).

Among the main concepts adopted in UFO-L, we highlight:

(i) `Agents` play `legal roles` in a `legal relation`. `Legal roles` are specializations of `social roles` and, therefore, they are sortals, anti-rigid and relationally dependent types. For example, in a consumer relation model, `agents` play the `legal roles` of buyer and seller in the buying and selling relation;

(ii) `Legal relators` are relational `moments` existentially dependent on other individuals who perform `legal roles`. A specialization of a `legal relator` is the `right-duty relator`. “A right–duty legal relator uses the legal relation right–duty (correlative) to bind right holder and duty holder. A right holder is someone who has a right to something against a duty holder (e.g. a citizen as right holder has a right to vote against the state as duty holder). A duty holder is someone who has the duty to materialize the right of a right holder” (GRIFFO *et al.*, 2015).

This is an on-going research. Here, we present a fragment of the domain ontology. A small fragment of the complex and huge Law domain was chosen to demonstrate that the LawV is feasible. This fragment is depicted in Figure 39. Complementing the ontology diagram, a glossary of the ontology elements is described in Table 26.

---

[38] For more details concerning UFO-L, see (GRIFFO *et al.*, 2015) and (GRIFFO *et al.*, 2016).

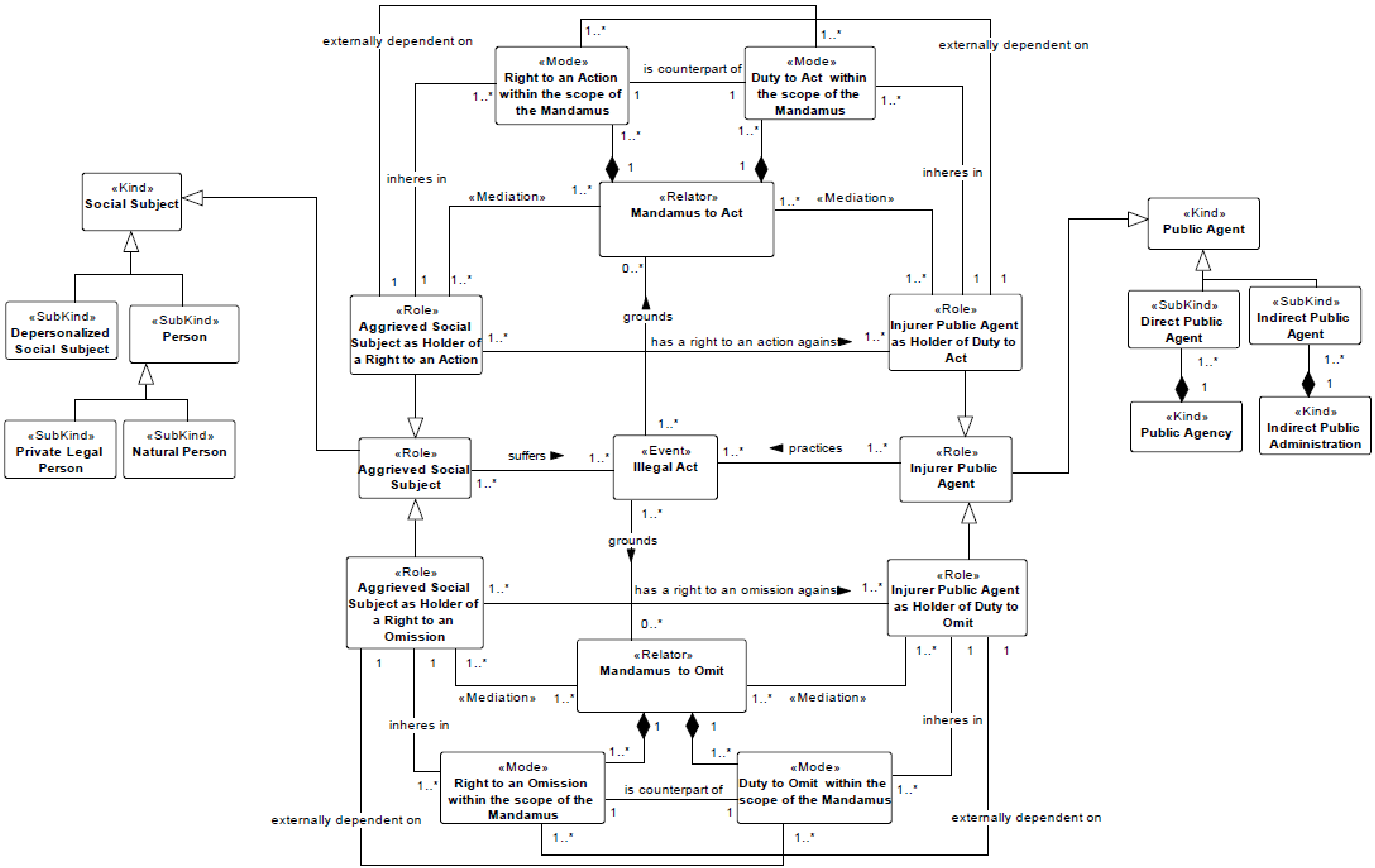


Figure 39: The reference ontology of Writ of Represssive Mandamus

The domain ontology developed is based on the Brazilian Constitutional Theory, specifically the theory on the Writ of Repressive Mandamus. A writ of mandamus can be summarized as follows: it is a constitutional instrument prescribed in Article 5, paragraph LXIX of the Brazilian Federal Constitution, which aims at the protection of clear and perfect right, individual and collective, not covered by *habeas corpus*[39] or *habeas data*[40] when the person responsible for commiting the illegal act, illegal threat or power abuse is a public authority or agent of a legal entity in the exercise of activities of the Public Power. Another disseminated definition of mandamus is given in (MILLER *et al.*, 2010): Mandamus is a judicial remedy which is in the form of an order from a superior court to any government, subordinate court, corporation or public authority to do or forbear from doing some specific act which that body is obliged under law to do or refrain from doing, as the case may be, and which is in the nature of public duty and in certain cases of a statutory duty. For example, in a public agency, the chief of a department hindered one of his subordinate to schedule her vacation. The subordinate employee requested a writ of mandamus to guarantee her right to go out on vacation, in accordance with the Brazilian law number 8112, article 77. So, the chief was obligated to schedule the subordinate employee vacation.

A complementary concept to mandamus is the clear and perfect right – the injury to this right justifies the use of a mandamus. The clear and perfect right is a right proven by unequivocal documentation, that is, that does not need to be proven. For example, the right to life is prescribed in the Brazilian Federal Constitution. The Law does not need to be proven, only its presentation or mention is enough. This concept is indirectly represented in the ontology, through the concepts of right to an action / omission and duty to act / omit, which are the remedies to correct the injured clear and perfect right.

Besides mandamus and clear and perfect right, other core concepts in the mandamus domain are: illegal act, aggrieved social subject, injurer public agent. They are defined in Table 26.

---

[39] *Habeas corpus* means “right to mobility”.

[40] *Habeas data* means “right to access and correct information that is in a public organization”.

Table 26: Glossary of the ontology of Writ of Repressive Mandamus

Ontology Element	Description
Mandamus to Act	The legal right-duty to an action relation reified through a `relator`. The legal characteristics inherent in the roles involved in the relation are parts of this entity, that is, they are the modes identified in the ontology.
Mandamus to Omit	The legal right-duty to an omission relation reified through a `relator`. The legal characteristics of the corresponding qua individuals are present here.
Illegal Act	The act practiced or ordered for the execution or non-execution of an illegality that damages the clear and perfect right of an individual. It is practiced by people who exercise public power directly or by delegation (public agents). Illegal act grounds the right-duty relation in a mandamus. Thus, an illegal act is a public act practiced by the public agent in the exercise of a public function. The illegal act can be either commissive (action) or omissive (omission), hence it can give rise to a writ of mandamus for an action or an omission. Example: The rejection of the public employee legal vacations by her chief.
Aggrieved Social Subject	The Social Subject found aggrieved who has the clear and perfect right. S/he is the subject (or its representative) who has had the right injured or threatened by and illegal public act. They can be active subjects: the universals or depersonalized entities, the private juridical person, the natural person and depersonalized public bodies, but endowed with procedural capacity. It is possible to have several active subjects on a writ of mandamus. For example: The public employee who has had her legal vacations rejected.
Injurer Public Agent	The Public Agent found injurer who practiced, threatened to practice or ordered the execution or non-execution of an illegal act that damages the clear and perfect right of the active subject. The following may be injurer public agents: subjects that are part of the Direct Public Administration (the Powers of the Union, of the States and of the Municipalities) and of the Indirect Public Administration (autarchies, foundations, public companies and mixed companies exercising public services), natural persons or legal entities with delegation of public powers (concessionaires and licensees of public services and utilities). For example: The chief who rejected the public employee legal vacations.
Right to an Action within the Scope of the Mandamus	It is the right to an action on a writ of mandamus. This right is an intrinsic element to the subject who has suffered the unlawful act and it is related to the duty to act, intrinsic element to the subject who practiced the illegal act (the public agent). In other words, this is the right to a public agent´s action towards the aggrieved social subject. For example: The right of the public employee to that the chief grant the legal vacations.
Right to an Omission within the Scope of the Mandamus	It is the right to an omission in a writ of mandamus. This right is an intrinsic element to the subject who has suffered the illegal act and it is related to the duty to omit, element intrinsic to the subject who practiced the illegal act (the public agent). In other words, this is the right to a public agent´s omission towards the aggrieved social subject. For example: the right of the driver to that the Chief of the State Traffic

	Department do not charge a fee to examine the application for annulment of the traffic violation.
Duty to Act within the Scope of the Mandamus	It is the obligation of the subject who promoted the illegal act to do something in face of the subject who suffered the illegal act. It is the duty that, by judicial order, must be fulfilled to remedy the injury to the right. In other words, this is the duty to an action of the injurer public agent towards the aggrieved social subject. For example: the duty of the chief to grant the legal vacations to the public employee.
Duty to Omit within the Scope of the Mandamus	It is the obligation of the subject who promoted the illegal act to stop doing something in face of the subject who suffered the illegal act. It is the duty to omit that, by judicial order, must be fulfilled to remedy the injury to the right. In other words, this is the duty to an omission of the injurer public agent towards the aggrieved social subject. For example: the duty of the State Traffic Department do not charge a fee to examine the driver´s application for annulment of the traffic violation
Public Agent	Any person who belongs to the structure of the Direct and Indirect Public Administration. These people exercise the will of the State and may not have a professional relationship with the State (eg, politicians), or they have some professional relationship with the State (eg civil servants, public servants, public employees, temporary employees), or they are individuals in collaboration with the State (e.g., the director of a public services concessionnaire).
Direct Public Agent	The public agent of the Direct Public Administration (Union, States and Municipalities). For example: the delegate, the fireman, the manager of a public bank.
Indirect Public Agent	The public agent of the Indirect Public Administration, that are, agents of municipalities, foundations, public companies and mixed societies exercising public services, and private natural or juridical people delegated from public powers in concessionaires and licensees of public services and utilities. For example: the director of a private school, the director of a public services concessionnaire.
Public Agency	The Direct Public Administration comprises parts called organizational units, which are instruments of the State will, deprived, as a rule, of legal personality. Direct public agents compose these organizational units. Example: the court of justice, the Presidency of the Republic, the State Department of Justice.
Indirect Public Administration	Entities with legal personality that compose the Public Administration, such as: public or private universities, public foundations, public enterprises, semi-public corporations; regulatory agencies.
Injurer Public Agent as Holder of Duty to Act	The role played by an authority who practiced an illegal act and therefore violated a right. Because of this, the authority who coerces holds the position of "must to do" some action during a writ of mandamus. Examples: the secretary of justice who denies legal holidays to the public employee; The police commander who denies the registration of candidates for reasons not prescribed by law.
Injurer Public Agent as Holder of Duty to Omit	The role played by an authority who practiced an illegal act and therefore violated a right. Because of this, the authority who coerces holds the position of "must not to do" some action during a writ of mandamus. Example: the director of a public company that imposes not negociated clause to the employess.

Social Subject	Every physical or legal person, native or foreigner, domiciled or not in Brazil and the universality (entities not-personalized – entities which are classified neither as natural person nor as a legal person. For example, the assets which form an inheritance).
Depersonalized Social Subject	Entities that do not have legal personality[41], but which, nonetheless, have their rights guaranteed by the Law, as examples: the estate and the bankrupt estate. Not-personalized public agencies are not included.
Person	Any entity capable of rights and duties in the civil order.
Private Legal Person	The private business entities with legal personality, created with a specific finality and governed by their own rules, that is, entities such as corporations and non-government organization, which are treated by Law similarly to natural persons.
Natural Person	Any individual (a human being) who holds rights and obligations.
Aggrieved Social Subject as Holder of Right to an Action	The individual that suffered injury in his/hers clear and perfect right or s/he represents the individuals who suffered the injury. This individual holds a right of action in face of the authority who coerces. Examples: the public employee who has denied his/hers request for legal holidays; The candidate who had his / her registration rejected in public tender for an illegal reason; The child who was denied his/her enrollment in a day care.
Aggrieved Social Subject as Holder of Right to an Omission	The individual that suffered injury in his/hers clear and perfect right or s/he represents the individuals who suffered the injury. This individual holds a right of omission in face of the authority who coerces. Example: the director of a public company that convenes an ordinary general meeting contrary to the statute; The rector of a public university that hires employees without public tender; The head of a public office that applies sanctions not prescribed by law to employees or citizens.
has a right to an action against	This legal relation is correlata to the relation "has a duty to act in face of".
has a right to an omission against	This legal relation is correlata to the relation "has a duty to omit in face of".
Is counterpart of	It is the relation that correlates a right to a duty, and vice versa.
Grounds	It is the relation that bounds an illegal act to the granted mandamus.
Suffers	It is the relation that bounds an illect act to the aggrieved social subject.
Practices	It is the relation that bounds an illect act to the injurer public agent.
Inheres in, externally dependent on, mediation	They are mandatory relations inherited from UFO-L patterns.

[41] The capacity of a person to have rights and obligations in Law domain.

In this case study, the cut-off of the domain that is apllied is the approval of the mandamus request (perspective of the person who had the right supposedly harmed). Other cut-offs made in the domain are:

- In the illegal act: In the field of the writ of mandamus, the public act is specialized in: (i) Illegal act; (ii) Legal act. The illegal act was not specialized in omissive and commissive acts. Only the illegal act is represented;
- In the active subject (social subject): the active subject Depersonalized Public Agency was not specialized. Universalities were named Depersonalized Social Subjects and people were specialized in Private Legal Person and Natural Person. Also, for the sake of simplicity, the composition relation between Natural Person and the Public Agency or Indirect Public Administration was not created, nor was the relation that transforms Natural Person into a Public Agent. This last relation is the representation of the process of investiture of public position, public contracting, delegation of public function and other modalities of exercise of the public function;
- Within the deadline of the writ of mandamus: as only the approval of the writ of mandamus is represented, the representation assumes that the deadline is satisfied, and concepts related to the deadline are not represented;
- In mandamus types: the ontology represents the type of repressive writ of mandamus that will order an obligation to do (right to an action) or an obligation of not to do (right to an omission). The type of preventive writ of mandamus (against threat of injury) is not represented in the ontology.

LawV can be used by lawyers in constructing the scenario of a possible writ of mandamus and in verifying the requirements for the impetration of the writ, as well as by magistrates in the construction of the scenario described in the records for understanding the whole situation and verification of requirements. Also, it can be applied as a method of learning in the discipline of Constitutional Law.

5.2.2.2 Design of the LawV Concrete Syntax

We organized this section in two sub-sections: In Section 5.2.2.2.1 we justify the choice of the problem domain applied, also indicating some considerations we should do in PoNTO-S approach due to two UFO concepts not dealt in the current release: `mode` and `event`; Finally, in Section 5.2.2.2.2, we described the application of PoNTO-S design process to generate the LawV concrete syntax.

5.2.2.2.1 Adjustments Required in PoNTO-S Design Process Due to Demands of LawV Abstract Syntax

We are aware that we have decided to use as domain ontology in this case study an ontology that does not fully satisfy the requirements of the current version of PoNTO-S. However, the Writ of Repressive Mandamus ontology is a real ontology in a complex domain - unlike the other empirical studies performed to evaluate PoN-S and PoNTO-S, which have resorted to simpler / less complex and smaller domains. What we intended in the current study was precisely to test the limits of the approach and use the case study to reflect on how the approach still needs to evolve.

In this sense, we had to make decisions even previously to the execution of PoNTO-S. This is because, as we depicted in Figure 39, two constructs used in the domain ontology do not have ontological guidelines associated: `modes` and `event` - it is worth mentioning that `event` is a concept of UFO-B, not yet contemplated in an OntoUML release. We have discussed the possibility of altering the domain ontology itself, replacing the stereotypes of `modes` and `events` by `kinds`, due to a connection that exists in the definition of such concepts. However, we discarded this hypothesis, since this would require that the developed ontology move away from its base (UFO-L) and it would mask the representation of concepts whose stereotypes have been purposely altered.

<u>Adapted Ontological Guideline to Deal with `Modes`</u>

Apparently, `mode` is not so applied as other OntoUML constructs (see Table 29 in Appendix C). So, initally, we had decided not to develop an ontological guideline for it in the first PoNTO-S release.

A `mode` is an intrinsic moment individual which is not a `quality`. `Modes` as much as `substantial universals` can be conceptualized in terms of multiple separable quality dimensions. Examples include beliefs, desires, intentions, perceptions, symptoms, skills. Like substantials, modes can bear other moments, and each of these moments can be qualities referring to separable quality dimensions. However, since they are moments, differently from substantials, modes inhere necessarily in some bearer. A special type of mode that is of interest in this case study is the externally dependent mode. Externally dependent modes are individual modes that inhere in a single individual but that are existentially dependent on (possibly a multitude of) other individuals that are independent of their bearers (GUIZZARDI, 2005a).

The fact that modes compose relators in the Writ of Repressive Mandamus ontology has also led us to resort to relations research to obtain support to the ontological guideline that we are delimiting. Guarino and Guizzadi (GUARINO; GUIZZARDI, 2015)(GUARINO; GUIZZARDI, 2016) argue that the distinctions between kinds, phases and roles should be applied to all endurants and not just object kinds. Thus, we could treat modes as modes-kinds, without losing the notion that modes are relational concepts (they are moments) and they will probably appear always participating in relations.

So, we decided to use the kind ontological guideline as a basis for dealing with mode construct, that is, a mode will also be represented through a shape. The differentiation of constructs in their definitions - the fact that modes inhere necessarily in some bearer - can be used to suggest the necessary complementary representation. In this case, we choose to represent the inhere relation as position - the mode should be positioned near its bearer (the proximity of the symbols presupposes the connection between them), and size - the mode should be represented in a smaller size than the representation of its bearer (in this way turning easier to identify who is the bearer and what is carried). The externally dependent on situation should also be represented through position – the mode should be positioned distant of the individual it is existentially dependent on.

Adapted Ontological Guideline to Deal with Events

Illegal Act is a core concept in the Writ of Repressive Mandamus Ontology. So, it is necessary to establish a manner to represent the Event concept, another concept, not initially planned to be part of the PoNTO-S ontological guidelines, as it is an UFO-B concept.

`Endurants`, as for example, `kinds`, are said to be wholly present whenever they are present, that is, they *are in time*. `Perdurants` (`events`) are individuals composed of temporal parts, they *happen in time* in the sense that they extend in time accumulating temporal parts. So, the main distinction between `Perdurants` and `Endurants` are in terms of their behavior with respect time (GUIZZARDI *et al.*, 2008).

As in the current version of the Writ of Repressive Mandamus ontology we are not exploring the temporal properties of an Illegal Act (the only concept classified as `event`), we decided to recommend the `kind` ontological guideline application in case of an `event` concept, and there is no need to add a complementary treatment to distinguish between `events` and `kinds`.

5.2.2.2.2 Applying PoNTO-S

To establish the concrete syntax, the language designer followed the PoNTO-S design process described in Section 4.3, complemented by the instructions for language designers applying PoNTO-S (see Appendix B) and the adjustments cited in Section 5.2.2.2.1. As a previous preparation, she studied PoN theory and she has a solid background knowledge in ontological theories. Also, some discussions were done between the involved researchers concerning PoNTO-S, so that the language designer had an adequate knowledge on PoNTO-S before applying it.

She started the design process execution commenting that the lifecycle overview (see Figure 25) allowed her to have a good overview of the complete process, the groups of PoN principles, as well as the involved assets (inputs and outputs), so, she is aware of what was expected from her.

In the first phase performed, *Specify dialect(s)* (see Figure 26), the designer navigated through each activity described, consciously identifying and commenting on each input / output, taking into consideration the Cognitive Fit principle. The outputs identified were:

- Stakeholder profiles: There are two profiles for probable language users, the Law professional and the Computer Science professional. The former profile should be the focus, however, we should not disconsider that a Computer Science professional could also be called to develop models in LawV. Also, the Law professional profile can be specialized in different subprofiles – a lawer, a judge, a prosecutor – and these profiles have different demands.

  LawV is being developed to attend mainly Law professionals, not used to graphical modeling languages. As a safety measure, the designer decided to consider the profile of a typical Computer Science professional, used to conceptual modeling, therefore, used in applying UML.

  The designer decided that is not necessary to treat separately the distinguished Law professional subprofiles. In the *Validation* phase, it will be check if something different should be adopted;

- Modeling task: It is expected that LawV will be applied in both development and interpretation tasks. Law professionals can be involved in both tasks and Computer Science professionals would typically be restricted to development tasks. Mainly, instance elements diagrams will be worked;

- Problem domain characteristics: The representation domain is a fragment of the Legal domain in the Constitutional field, more specifically the writ of Mandamus. The core characteristic of this domain is the presence of specific relations between the involved entities in a mandamus – legal `relators`, legal `roles`, legal relations;

- Number of dialects: Taking into consideration mainly the stakeholder profiles, the designer decided that two dialects would be interesting, each one focusing on a profile.

  - Dialect for Law professionals – goal and directives: it should be focused on simplicity and ease to use, turning the language simpler and easier to apply by non-technical professionals, who are more used to work with textual representations. PoN principles to be focused on: Semantic Transparency, Graphic Economy, Dual Coding;

- Dialect for Computer Science professionals – goal and directives: it should be focused on the modeling tradition in that field (more specifically, the UML notation), considering that this is a representation that the majority of professional modelers are used to;
- In both dialects, it is not necessary to represent any abstract entity directly. The designer pointed out that, apparently, there is no situation in which an instance element needs to be associated with a construct that is not yet fully specialized. The defined abstract types in the domain ontology are just for organizing more specialized types.

The core phase is the *Implement dialect(s)* (see Figure 27), mainly the *Define dialect symbol set* compound-activity (see Figure 28). Firstly, the dialect for Computer Science professional was established. The symbol set for this dialect was quickly decided: it is basically the same of UML object diagram for entities and relations instances (rectangles and lines. The lines should have direction when judge this would help to interpret the diagram). The identification of each concept (an entity or a relation) should occur through stereotype labeling. In the case of entity, besides the name of the instance element also the identification of the type should be described. This is in accordance to the dialect goal. The designer pointed out that: (i) If a Computer Science professional is not familiar with UML notation, s/he can simply resort to dialect 2; (ii) If a Computer Science professional is used to UML notation s/he could feel more confortable and s/he could be more productive working with dialect 1. Due to this decision, the designer noticed that it is not necessary to perform the *Implement dialect(s)* phase for dialect 1. However, she decided to execute the *Validation* phase together for both dialects.

Posteriorly, the dialect for Law professional was designed. In this case, the design process was applied in detail.

Concerning the activities *Analyze abstract syntax VML* and *Analyze dialect set VML*, the designer felt confortable that she knows deeply the domain ontology, given that she was one of its proponents. Nevertheless, she reviewed the ontology just to be sure. When working with <u>Act</u> (initially possessing a generalization involving the subtypes <u>Omissive Act</u> and <u>Comissive Act</u>) the designer noticed that instead of the grouping Act + Omissive Act + Comissive Act, for the domain under study, only the entity type Illegal Act would be necessary, replacing the three concepts initially visualized in the domain ontology. This could be an indication that during the concrete syntax design, the designer should reflect in-depth on the domain and s/he could notice that some adjustments in the abstract syntax are necessary.

To initialize the *Define dialect symbol set activity*, the designer decided that each symbol would be hand drawn. However, when reasoning in the first symbol, she changed her mind and in the rest of the process she searched for symbols on the Internet, complementing the creation of symbols with using a simple image editor for drawing some symbols when she did not find what she was looking for on the Internet.

The designer was constantly aware of the ontological guidelines and that the symbol set should be defined according to them. Mainly the guidelines expressed in Table 28 and in Section 5.2.2.2.1. Regarding an order to choose a model element, even being aware of the ontological guideline concerning this aspect, she opted to be guided mainly by groups of elements and to firstly represent concepts which symbols she could imagine more naturally. After establishing the more natural symbols, she turned her attention to more difficult representation – not so natural symbols.

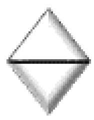
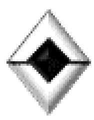
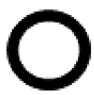
Another highlight made by the designer is that during the design process, the ontological guidelines were of significant help. However, even with this help, the designing task is difficult, because the major difficulty is to identify a symbol, mainly when there is no obvious symbol or standard icon to be used. For example, when reasoning about the Natural Person representation, the stickman index came to mind, but how to represent a Depersonalized Social Subject? Both examples are cases of `subKinds`, so, the designer already decided that they should be represented through the shape visual variable, but even in this case, the activity of establishing a symbol remains difficult. This difficulty should be caused by the domain. Regularly, the designer was recurring to the Internet, seeking for an image that she could associate to a model element. However, rarely, a suitable image was returned in these Internet searches. So, she claims that several Law concepts do not have a standard symbol for representation. On the contrary, usually there is no established representation at all for them. Sometimes, there is an abstract form, as a rectangle connected to text, but this basic use of graphics is not so efficient as a semantically meaningful symbol.

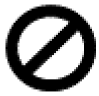
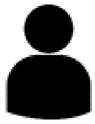
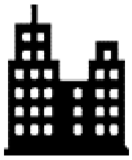
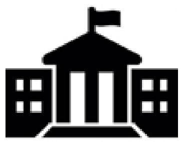
The designer was always conscious of the peculiarities of the Legal domain. For each symbol she was establishing she tried to connect it to the domain and she reasoned on how easily the language user would associate the symbol to its meaning (a Semantic Transparency principle concern). Even more, the domain constantly interferes in the order the model elements were chosen to be represented – close connected model elements are represented in tandem. Futhermore, easy model elements to represent are chosen first to be represented.

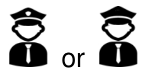
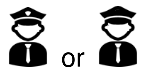
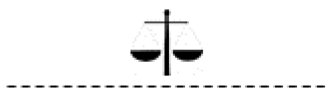
Since the first performance of the *Choose a model element to be represented* activity, the designer identified that it would be difficult to separate some modeling elements and that it is more natural to work in groups of elements. Probably, as a consequence of this, the *Associate a symbol to the model element* activity, and the *Relate new symbol to already defined symbols* activity were also performed together as a single activity. Usually, when establishing a symbol, the designer kept in mind the other concepts directly afected by the current model element under design. For example, when reasoning about rights symbols, the designer was worried with how to represent the duties symbols, because one should be the counterpart of the other symbol represented.

The symbol set defined is these activities are described in Table 27. After the table, we describe the design rationale adopted when establishing the symbols.

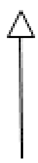
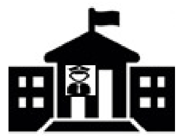
Table 27: Symbol set of dialect for Law professionals

Concept	OntoUML construct	Symbol	Applied visual variables
Entity type			
Right to an Omission within the Scope of the Mandamus	`Mode`		Shape Size – small symbol
Right to an Action within the Scope of the Mandamus	`Mode`		Shape – adding of detail Size – small symbol
Duty to Act within the Scope of the Mandamus	`Mode`		Shape Size – small symbol

Duty to Omit within the Scope of the Mandamus	Mode		Shape – adding of detail Size – small symbol
Social Subject	Kind	Abstract entity type – not represented directly	-
Person	Subkind	Abstract entity type – not represented directly	-
Natural Person	Subkind		Shape
Private Legal Person	Subkind		Shape
Depersonalized Social Subject	Subkind		Shape – update of detail
Public Agent	Kind	Abstract entity type – not represented directly	-
Direct Public Agent	Subkind		Shape – adding of detail
Indirect Public Agent	Subkind		Shape – update of detail
Public Agency	Kind		Shape
Indirect Public Administration	Kind		Shape – update of detail
Aggrieved Social Subject	Role	Abstract entity type – not represented directly	-
Injurer Public Agent	Role	Abstract entity type – not represented directly	-
Aggrieved Social Subject Right to an Action Holder	Role	 or  or 	Shape + size – duplication of the supertype entity in a

			reduced size
Aggrieved Social Subject Right to an Omission Holder	`Role`		Shape + size – duplication of the supertype entity in a reduced size
Injurer Public Agent as Holder of Duty to Act	`Role`		Shape + size – duplication of the supertype entity in a reduced size
Injurer Public Agent as Holder of Duty to Omit	`Role`		Shape + size – duplication of the supertype entity in a reduced size
Illegal Act	`Event`		Shape
Mandamus to Act	`Relator`	<<label>>	Shape + texture (applied in the border) Support of text Optional: texture and color to differentiate among several instances of both relators
Mandamus to Omit	`Relator`	<<label>>	
Relation			
is counterpart of[42]	`Formal`		Shape + position (connecting a Right to a Duty) + texture
Practices	Association		Shape + color + position
Suffers	Association		Shape + color + position
grounds (there are two relations using this name. They are treated as the same relation)	Association		Shape + color + position + texture
has a right to an action against / has a right to an omission agains	`Material`	Indirectly represented throught the representation of `relator` + `roles` +	-

[42] There are two relations named like this. The is counterpart of relations between Rights and Duties were visualized during Validation phase.

		`mediation` relation	
inheres in	`Characterization`	There are four inheres in relation connecting `modes` and `roles`. They are represented in the same manner	Position – the `mode` representation should be placed close to the `role` represented which it is inherent.
externally dependent on	`Characterization`	There are four externally dependent on relation connecting `modes` and `roles`. They are represented in the same manner	Position – the `mode` representation should be placed distant to the `role` represented which it is correlate.
There are 4 mediation relations. They are represented similarly	`Mediation`	Attchament of the involved role symbol in the border of the involved relator symbol	Position
gs-public agent, gs-social subject, gs-person	Generalization	Indirectly represented. The subtypes involved in each generalization relation have a clear connection between each other – There is a basic shape and addition of details to this basic shape	-
gs-role-aggrieved-social-subject, gs-role-injurer-public-agent	Generalization		Shape + position + color (the same color applied to the `relator` and replied to the `roles` involved in this relation)
componentOf public agency	ComponentOf		Position – spatial inclusion + size – reduction of the element symbol
componentOf indirect public administration	ComponentOf		Position – spatial inclusion + size – reduction of the element symbol

Regarding the sequence of chosen model elements to represent[43] and the design rationale adopted when establishing their representation:

- Right to an Action within the Scope of the Mandamus, Right to an Omission within the Scope of the Mandamus concepts (`modes`): they should be

[43] The sequence which the model elements are presented by the markers reflects the chosen sequence for representation adopted by the designer.

represented as shapes. They are something that a Social Subject wins – a type of conquer. The idea of a diamond arose. The representation of the Right to an Omission within the Scope of the Mandamus was the basic symbol while an adding of detail to the diamond shape is established to the other `mode` (Right to an Action within the Scope of the Mandamus);

- Duty to Act within the Scope of the Mandamus, Duty to Omit within the Scope of the Mandamus concepts (`modes`): How to represent the counterpart of a Right? This required substantial consideration from the part of the designer. In the end, she decided to apply the shape of a circle. It is interesting that to identify these symbols the base was the more detailed symbol, Duty to Omit within the Scope of the Mandamus, recurring to the traffic sign for prohibition (a line crossing the circle). After this, the Duty to Act within the Scope of the Mandamus was simply the circle without the crossing line;
- is counterpart of (an association): This relation expresses a strong connection between the concepts of Rights and Duties, as `modes` that are part of a `relator` and that are co-dependent on each other. As a matter of fact, only after defining a symbol to represent this relation, it was possible to be certain about the representation of Righs and Duties. The first idea was to represent these two concepts as complementary shapes to identify their correlation, but that idea was discarded and the designer decided to represent the relation through a line connecting the `modes` representation and a decorative symbol attached to the line. This relation derived from the `correlates` relation of UFO-L. Correlation gives idea of equilibrium, and because of this, the symbol of the balance was chosen. Also, the balance is a symbol found in the Law domain.

Rights, Duties and their connection as counterparts of each one, are the core concepts of LawV. That is why, they were visualized first, even not being the simpler concepts. After them, the simpler concepts were represented:

Additional central concepts in the LawV domain are represented through its `relators`:

- Mandamus to Act, Mandamus to Omit (`relators`): the representation indicated in the correspondent ontological guideline was adopted. The designer decided to apply a rectangle to represent `relators`. This shape is reinforced by a bold borderline (texture visual variable). As secondary notation, she decided that color could be used to differentiate between different instances of these `relators`. Keeping the dialect directive of applying the Graphic Economy principle, the designer decided that both `relators` should be represented using the rectangle shape. Text should be applied to identify each `relator`;

- Natural Person, Private Legal Person, Depersonalized Social Subject (`subkinds` of a same `kind`) – It was easy for the designer using an index of person to represent the Natural Person entity type. Also, Private Legal Person was associated to the index of a building, as usually this concept is associated to an enterprise. The difficulty arose when representing the Depersonalized Social Subject. How to represent a concept that is the complement of Person? A Social Subject without personality. After a while, the designer established a parcial replication of the Person representation – to give the impression that Depersonalized Social Subject is an incomplete Person representation, adding a cloud symbol as detail, because a cloud is something without a defined contour, something difficult to classify because it does not have a typical format. Social Subject and Person are not represented because they are abstract types;
- Direct Public Agent, Indirect Public Agent (`subkinds` of a same `kind`) – Public Agent is a human person as Natural Person, but they are not directly connected. So, the designer decided to apply a similar but different symbol of the adopted to Natural Person (a representation for human person). As public agents are workers of the Law, she reasoned that adding detail to the basic symbol would be enough to identify each concept. A tie and cap are the details. A coat of arms in the cap is used to differentiate between the Direct Public Agent (an official position) and the Indirect Public Agent (not official position). The designer was in doubt if the discriminability between the symbols was enough. This is an issue she intends to analyze during the Validation phase. As an abstract type, the Public Agent concept is not represented directly.

In the first attempt, the same basic symbol of Natural Person was applied. However, during the validation phase, when the designer was more aware that Natural Person and Public Agent are different `kinds`, she decided to update the basic symbol of Public Agent, establishing a more different representation;

- Public Agency, Indirect Public Administration (cases of `kinds`) – These are concepts of places, so the representation of a building was applied, reminding of a common Justice building representation (a type of Roman forum). As in Public Agent, a shape detail was chosen to differentiate between these two types. In this case, Public Agency, being an official place, has been decorated by a flag.

  There was doubt if these concepts should be represented as subtypes of a same supertype. However, the designer decided it is not necessary to represent this connection. At maximum, an abstract supertype could be applied.

  Also, the designer was in doubt if the discriminability between the symbols was enough. This is an issue she intends to analyze during the Validation phase;

- Aggrieved Social Subject, Aggrieved Social Subject as Holder of a Right to an Action, Aggrieved Social Subject as Holder of a Right to an Omission (`roles`) – As `roles`, they should be represented as miniatures of the `sortal` representation that gives rise to them, that is, the possible representations for Social Subjects in this case. We should notice that the only difference between the `roles` representation would be the position (the `relator` representation near which each one will be placed). As Aggrieved Social Subject is an abstract type, the designer decided not to represent it;

- Injurer Public Agent, Injurer Public Agent as Holder of Duty to Act, Injurer Public Agent as Holder of Duty to Omit (`roles`) – As `roles`, they should be represented as miniatures of the `sortal` representation that gives rise to them, that is, the possible representations for Public Agent in this case. We should notice that the only difference between the `roles` representation would be the position (the `relator` representation near which each one will be placed). As Injurer Public Agent is an abstract type, the designer decided not to represent it;

- Illegal Act (`Event`) – After several ideas discarded, the designer opted for a composite symbol: a shape of an official paper (reminding the idea of law) crossed by a red "x" (reminding the idea of prohibition). Initially, Illegal Act was considered as a `kind`. However, even after its update to an `event`, the designer followed the adjusted ontological guideline and handled with the concept as a `kind`. She commented that she did not notice any need to represent it differently because it is an `event`. In the validation phase, she intends to check this issue;
- suffers, grounds, practices (regular associations) – As regular associations, the designer decided to apply the regular representation for relation, the shape of a line and a triangle indicating the direction of the relation (so, it is a composite symbol). However, a doubt arose: how to differentiate among these regular associations and also among other relations represented as lines? The ontological guidelines recommend the use of shape, position and texture, but the designer decided that they were not enough. Instead of this, she applied color to differentiate between the relations – This suggests it could be interesting to add color visual variable in the set of visual variables recommended for representing relations. Also, as suffers and practices are closer connected between each other then grounds, to differentiate the latter of the others, a different value of texture (use of a dashed line) was considered;
- `Mediation` relations – There are four `mediation` relations in LawV. The designer decided to adopt the correspondent ontological guideline of PoNTO-S, that is, each `mediation` relation is represented as the attachment of the `role` representation to the `relator` representation;
- Generalization relations other than those involving `roles` – This type of relation is not directly represented. However, there is an indirect representation of them, due to the fact that each connected subtype is represented considering a basic shape and some adding of detail to this shape, as for example Direct Public Agent (its representation adds a shape

detail – the coat arms – to the basic shape) and Indirect Public Agent (the basic shape);

- Generalizations involving `roles` – Unlike the other generalizations, the identification in a diagram of the supertype of each `role` is recommended to be clearly established. As recommended in the correspondent ontological guidelines, the designer applied the traditional symbol of generalization (a line attached to a transparent triangle containing black borders) to represent that a Social Subject plays the role of Aggrieved Social Subject, as well as a Public Agent plays the role of Injurer Public Agent. The color of the `relator` that mediates the `roles` should be replied in this relation representation;

- `ComponentOf` relations involving Public Agency and Indirect Public Administration – In both relations, the designer adopted the recommendation of the correspondent ontological guideline, recurring to spatial inclusion to represent the `componentOf` relation. The chosen representation is suitable to represent the generic dependence and non-shareability metaproperties, both properties of the `componentOf` relations presented in the domain ontology.

  The choice of the symbol of the whole element (Public Agency and Indirect Public Administration) made it difficult to represent the relation with the part ement (Direct Public Agent and Indirect Public Agent, respectively) as spatial inclusion. There was doubt if it would be better in this case not to follow the recommendation of PoNTO-S, using the traditional representation of relation (a line connecting the elements) connecting the whole and part elements. After some reflection, the designer decided to keep the recommendation of PoNTO-S as well as the symbols chosen.

  A suggestion that resulted from the establishment of this symbol is to deepen the study on the representation of meronymic relations, especially in a situation which the quantity of part-elements instances involved in the relation is large;

- Relations inheres in and externally dependent on - they are relations that occur four times each in the ontology and they are represented in the same way in such repetitions. In addition, they are complementary relations, and thus, the designer decided directly interrelated representations to represent such relations - the spatial position, by doing this through proximity (closer and more distant values). However, there is doubt whether the externally dependent on relation will become clear to be presumed in a diagram interpretation. The experiment is expected to demonstrate this. As a result of this representation, it was noticed that they are directly linked to the `relator` representation – it would be easier if it was represented together: the `relator`, the `roles` involved, the `modes` involved, the links between `relator` and `roles`, and the links between `roles` and `modes`. After establishing the representation of the `relator` and the `roles`, their connections, after establishing the representation of the `modes`, then the designer decided to place each representation of `mode` next to the `role` in which each one is inherent, she noticed that automatically the `modes` would already be far from the `role` of which they are correlated.

After establishing the symbol set, the designer analyzed if it would be necessary to perform the *Apply support principles* activity. She concluded that during the previous task (defining the dialect symbol set) she already kept the support principles in mind, so, it would not be necessary to reavaluate them once more.

Regarding the *Identify ways to manage complexity in the dialect* activity, responsible for the creation of representation strategies for managing model complexity, the designer opted for not develop any representation strategy – she believes that they would not be necessary, because the resulting diagram would not be complex or large. The activities of *Evaluate complexity of symbol set*, *Integrate information from different diagrams* and *Apply support principles* were not cited.

Once the symbol set establishment and definition of representation strategies of models were done, the designer proceeded to the next phase, Validation, in order to evaluate the concrete syntax, following the activities proposed by PoNTO-S.

Concerning the activity *Check PoN´s principles conformance*:

- Dialect 1 (for Computer Science professionals): The designer was aware that there are publications available in the field that analyze the UML diagram family, including class diagram, as in (MOODY; HILLEGERSBERG, 2009). Moody and Hillegersberg pointed out some problems found in the visual notation of UML diagrams. They claim that these can impact the usability of

UML. However, considering that huge adoption of UML in modeling tasks, the designer opted to keep the dialect 1 without any change;

- Dialect 2 (for Law professionals): The symbol set was created according to the PoNTO-S recommendations. So, it is expected that the dialect satisfies the constraints posed by PoN. The main conclusions refering each PoN principles are:
  - Semiotic Clarity: All the entity types and relations are mapped to some symbol. Only abstract entities don´t have a symbol directly associated to them. So, the principle was satisfactorily attended;
  - Semantic Transparency: The designer intended to apply symbols commonly used in Law field, as the balance index, an image associated to Justice, to represent the is counterpart of relation, creating a balance between rights and duties. However, she noticed that there are not graphical symbols usually associated to Law concepts. Usually, the Law field uses text. So, the designer decided to establish easily mapped symbols, to make the learning of the dialect easier. Despite these considerations, the principle was satisfactorily attended;
  - Perceptual Discriminability: Observing Table 27, we can notice that connected concepts are easily identifiable, as adding of detail to a basic shape was the most common manner of establishing symbols – this allows to keep the similarity between concepts and it shows that there are differences between the concepts. The only problem identified here is concerning the representation of `relators`, because the two concepts have the same representation (a rectangle form). This decision was adopted benefiting Graphic Economy. Despite these considerations, the principle was satisfactorily attended;
  - Visual Expressiveness: The designer applied shape, color, position, size and texture visual variables, according to PoNTO-S recommendations. Color and shape are the most used visual variables;
  - Graphic Economy: This principle is indicated as a directive of the dialect. At some moments, the application of this principle is clear, as in the representation of both `relators` (the same symbol is used to represent

the two `relators`), even if this causes a symbol overload situation. Also, there are repeated relations, as in the cases of inheres in and externally dependent on. These relations connect different entities, but the relations have exactly the same meaning. So, the designer decided that is not necessary to represent the relation differently. In addition, the designer tried to keep the selected symbols as simple as possible, only putting the details deemed necessary;

- Dual Coding: Text plays an important role in the Legal field. According to PoNTO-S, the use of text should be a complement of graphical notation. In the case of the representation of `relators`, text is the only differentiation between the two entity types (the two `relators` of the domain have the same symbol associated to them). During the elaboration of diagrams, it is suggested that every entity instance has an associated label;
- Complexity Management: Firstly, none representation strategy is defined. The designer decided that during the planned empirical study it will be evaluated if some strategy is demanded. So far, the principle was considered but did not produce any result;
- Cognitive Integration: It applied the same observation made to Complexity Management principle;
- Cognitive Fit: There are two stakeholder profiles identified. The designer judged that this situation demands the creation of two dialects, each one specific for each stakeholder profile. So, again, the principle was satisfactorily attended.

The designer considered that the checking of PoN principles conformance is positive for her. She concluded that there is no need of changing in the defined symbol sets. We agreed with her. However, the next activities of the validation phase are necessary to assure this first impression.

Concerning the activities *Generate diagram instance* and *Evaluate diagram instance*:

The designer generated a model instance, which text is the following: "The Civil Policeman of the Espírito Santo State (someone named Dedier) is in probationary stage. He requested on February 1, 2017[44], an unpaid leave for dealing with matters of private interest (ULDMPI) to carry out the training course for a Federal Police position at the National Academy of Federal Police by reason of approval in a public tender. However, the chief of the Espirito Santo State Civil Police (ESSCP) denied the unpaid leave based on the Article 41, sole paragraph of the Complementary Law No. 46/94 that does not allow the granting of ULDMPI for public servers in a probation period. The act denying the license was published in the Official Journal on February 20, 2017. In disagreement with the decision, the public employee, filed a petition for writ of mandamus in time so that his right to access a public position was protected and corrected for the injury suffered. His request was granted and an order was given for the Chief of Police to grant the unpaid leave for dealing with matters of private interests for Dedier".

The instance element diagram produced using dialect 1 is depicted in Figure 40. Likewise, dialect 2 was used to represent the same model instance. Figure 41 contains the representation. Dialect 1 seems to produce a diagram highly dependent of text to be interpreted. On the other hand, dialect 2 results in a diagram more graphical (there is less dependency of text to be interpreted).

Next, we describe some claims identified by the designer when building the instance elements diagrams.

- At first, when executing the activity *Identify ways to manage model complexity*, the designer imagined that two abstraction levels representation would be required to represent a repressive mandamus situation, one more complete and the other simplified (with fewer concepts being visualized). However, when building the first instantiation, the designer realized that this representation was already simple enough, not needing an even more simplified view. Therefore, she returned to the activity in question and

[44] The dates are not real.

concluded that it is not necessary to develop any representation strategy for LawV in its current version;

- The designer decided that in the LawV manual the following recommendation will be written: the modeler should label some entities intances. The entities are: Social Subject, Public Agent, Public Agency, Mandamus, Illegal Act, Right, Duty. Also, the `relators` should be named, as previously commented. This should be done taking into account the high use of textual representation in the Legal domain tradition. The other concepts do not need textual support;
- Concerning recognition of the mandamus: at first, the designer believed that identify the name of the type and the instance elements would be necessary, as both `relators` of the ontology are represented in the same manner. However, during the activity she changed her mind. If a suitable label is chosen to a mandamus, this name would be enough to identify it as a mandamus to act or to omit;
- Observing the connected representation of the Public Agency, the Direct Public Agent, the `componentOf` relation between them, the generalization of Public Agent involving its `role`, and also the position of the labels associated to each one of these representations, the designer had doubts regarding these representations as a whole – if they are clear enough in interpretation tasks when placed together. However, she decided that will evaluate the suitability of the representations during an empirical study, hoping that the experiment probably give her a more suitable representation, if necessary so. Changes will be provided only if they are necessary (if the participants demonstrate difficulty when interpreting these group of representation).

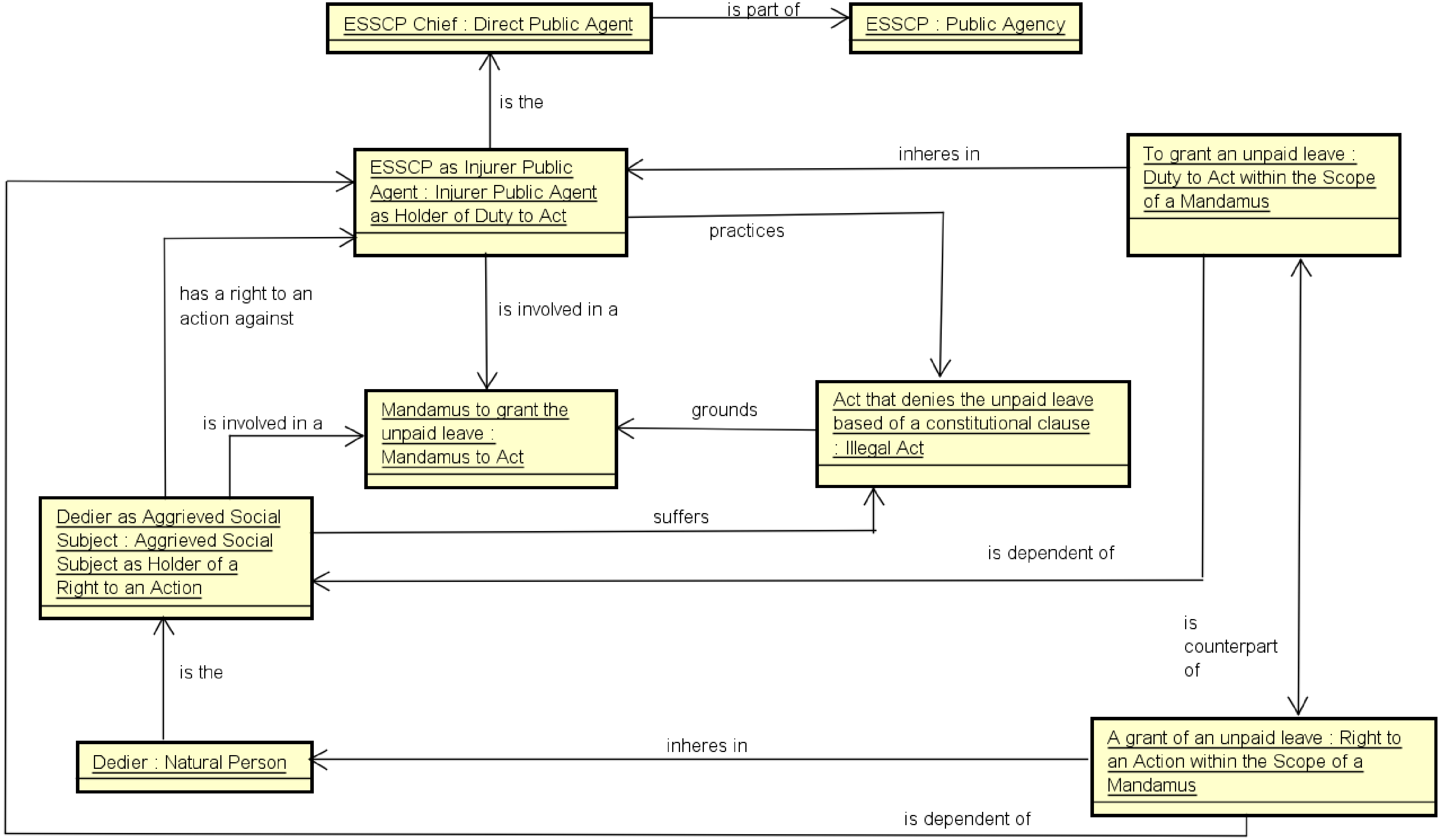


Figure 40: Instance elements diagram generated using dialect 1

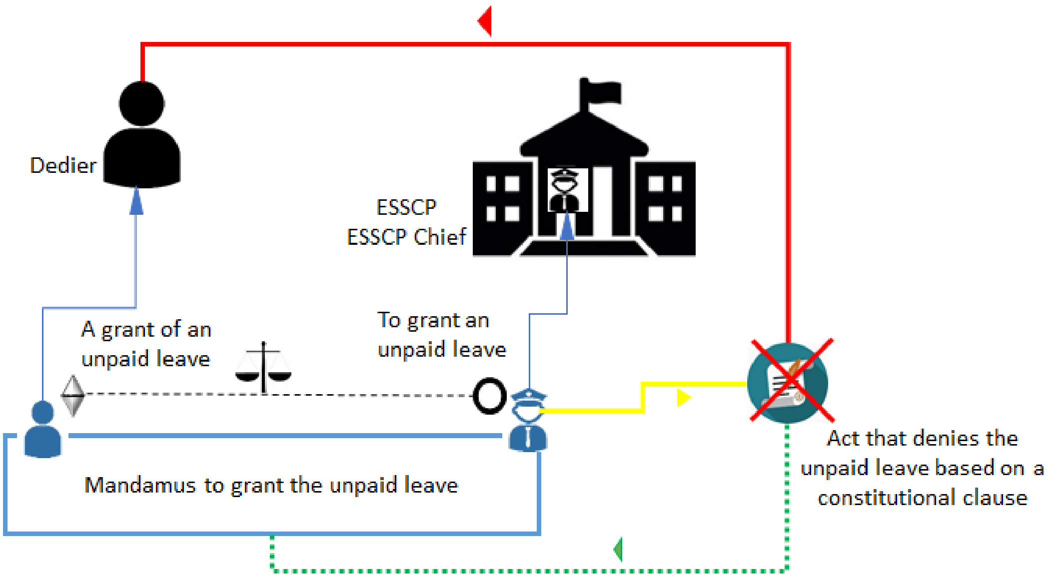


Figure 41: Instance elements diagram generated using dialect 2

The above observations are related to a single instance. The designer judged the obtained results enough to complete the activity. Some identified evidence generated a report (see Section 5.2.2.3) containing suggestions for changing PoNTO-S.

Up to this point the dialects evaluation by the designer was carried out. Although desirable and in planning stage, the application of empirical studies is an activity not yet performed. Anyway, the designer concluded that is worth to perform empirical studies for evaluating the use of the proposed dialects.

With regard to the *Apply empirical study* activity: Such activity is under planning. In principle, two empirical studies will be performed. Their goals will be: (i) Interpretation of mandamus writ instances developed using LawV compared to having semantically equivalent representations in NOMOS (SIENA, 2010) languages; (ii) Interpretation of mandamus writ instances using the LawVML language compared to having semantically equivalent representations only in natural language (text).

5.2.2.3 Final Considerations

The previous section presented the application of PoNTO-S in the viewpoint of the designer. Our viewpoint of the task is described next, including the main evidences we collected during the process.

Concerning the designer behavior during this application, the issues we highlight are:

- The designer has a solid knowledge in UFO and the domain conceptualization. She studied PoN, PoN-S and PoNTO-S before performing the design task. So, we consider she has a solid background knowledge and this situation helped a lot in the task;
- We asked the designer to think aloud during the design process in order for us to take notes. This gave us the information necessary to elaborate the report described in Section 5.2.2.2.2;
- Once the designer studied PoNTO-S, including the instructions (see Appendix B), when performing the design process, she was following the activity diagrams, without consult again the intructions, except the table containing the ontological guidelines (Table 28);
- It was a cyclical process, not only for the creation of dialects, but also for the creation of the symbol set within each dialect, and the regular checking of the domain ontology;
- The designer commented that she felt more secure about the notation she created since she was following a process and could adopt the guidelines for details. She commented that in a previous experience of creating the concrete syntax of a modeling language, she felt more insecure about the proposed language, since she had applied only her intuition to perform the task;
- At several moments, when the designer was in doubt about a decision (which symbol to apply, for example), she decided that this doubt could be solved during the validation phase. Indeed, during the validation phase, the concrete syntax and even the abstract syntax were refined;
- As a general impression, the designer claimed that the design process and guidelines (PoN principles and ontological guidelines) were helpful. However, it was an exhaustive process that demanded several days to be completed.

We agreed that the proposed dialects could be helpful for Law professionals, in accordance to the claims of the Visual Law field. Even if we are in doubt if dialect 1 could be helpful, we believe it is worth to test it in an empirical study. Concerning dialect 2, the application of PoN principles and the ontological guidelines are clearly identified in the dialect proposed (see Table 27). Finally, we believe that the remaining doubts identified by the designer could be solved through the empirical studies and we expected that these activies would result in an improved dialect.

Concerning the draft of new ontological guidelines:

- When we decided to use a domain ontology that contained the constructs of `modes` and `events`, both constructs that are not contemplated in the current release of PoNTO-S, we became more conscious of the need to develop a more complete set of ontological guidelines, covering not only UFO-A concepts, but also UFO-B and UFO-C. The more restricted the set of constructs we provide, the smaller the number of domains that PoNTO-S will be able to assist;
- On the other hand, the process of adapting ontological guidelines to `modes` and `events` was interesting, as we recurred to the existing guidelines in order to speed the process. We use the closeness of these concepts to the `kind` concept to define how they should be treated. This attitude - using existing guidelines to adapt new constructs - is probably the way out to address constructs not yet covered by PoNTO-S. It can be added as a recommendation in PoNTO-S. Of course, this only works if the new constructs have a definition that connects them to the constructs for which guidelines already exist, as was the case. The guidelines for `modes` and `events` are not complete yet, but what were drafted here can be used as a basis to complete these ontological guidelines.

As evidences of PoNTO-S application that resulted in suggestions for improvement in the approaches, or, at least, issues to receive an in-depth analysis:

- Applying PoNTO-S approach made the designer more aware of the impact of her decisions before the effective language use and some changes could be made before the language is made available for use. Thus, the language is made available for use with fewer flaws;

- According to the designer reasoning, PoNTO-S seems to have influenced the designer to be more conscious of the need to use the PoN principles and/or the ontological guidelines, as it was clear at several moments when she expressed the reasoning adopted;
- The addition of details in a basic symbol when estabilishing a new symbol is an easy way to work. Once the designer identified a suitable basic symbol, it seems more natural to identify the new symbols generated from it[45];
- The fact that no representation strategy was judged necessary would lead us to believe that we should reinforce the importance of complexity management in the development of diagrams - so these are activities that we should give more support in PoN-S and PoNTO -S;
- When drawing the concrete syntax, the designer acquired a higher understanding of the abstract syntax, which conducted her in doing updates in the domain ontology. It may be interesting that the design process reflects this possibility - the DSML design process should be a cyclical process, involving both abstract and concrete syntaxes. Several cycles may be required until the designer concludes that the language (and all its elements) is suitable for use;
- The representation of a whole-part relation as spatial inclusion ends up restricting the symbol that will be used to represent the whole element. Considering that an element is part of this type of relation is fundamental to easily choose the symbol. It may be interesting to have alternatives to represent whole-part relations[46]. The connection between the entities involved in the relation as well as the relation itself may interfere with the chosen symbols. In addition, it may be interesting to have more flexibility for representing this group of concepts. This conclusion was pointed out due to the representation of `componentOf` relation in the dialect 2. Thus, it is

---

[45] When the size and complexity of the modeling elements grow, this working style would become more complicated – to differentiate among a huge number of similar symbols.

[46] As for example, instead of using spatial inclusion to represent the whole-part relation through a line connecting the involved elements. In this way, there would be no interference in the choice of each symbol.

necessary to study more deeply the representation of meronimic relations, especially when the quantity of part elements instances is large;

- To update the ontological guideline regarding represention of relations. The designer chose to use the traditional representation of relation. As this representation was used several times, to help in differentiating between relations besides shape, position visual variables, the designer also applied texture. In addition, she added the possibility of using the color visual variable;
- About applying support principles – it sounds as a secondary activity. During the establishment of the symbol set, a designer aware of PoN has such principles in mind. A separated activity to apply them seems to be valid only in cases which the designer wants to be sure of his/her decision. It may be worth reviewing the design process to allow the designer to keep the supporting principles in mind during the symbol establishment;
- The necessary characteristics of the model instances (an input for a validation activity) should be better described. The designer had doubts how she should describe the model instance she used;
- It could be enlightening to identify if the diagrams at sentence level would be of type-level or instance-level elements (or even both) in the requirements of the language, as this could guide the decisions made by a language designer.

The issues listed here were not yet adopted in PoNTO-S. As future works we will do an in-depth analysis of these issues and evolve PoN-S and PoNTO-S to consider what we judge necessary in a next release.

On the LawV creation process: In addition to applying at least one empirical study to complete the design process, the designer, when reasoning on the domain ontology during the validation phase, identified the situation as possessing temporal characteristics. She decided to generate a new version of the ontology, more complete in this sense. This new version will be later worked, including a new application of PoNTO-S, giving rise to a new version of the language. The cut-off presented here was identified as the end of this temporal process, when the approval to generate the mandamus occurs.

Chapter 6. Final Considerations

This chapter contains our final considerations. It presents an overview of the conducted research (Section 6.1). It describes the research contributions (Section 6.2) and briefly discusses the difference between PoN-S or PoNTO-S (Section 6.3). Finally, it describes limitations and future research directions (Section 6.4).

6.1 Overview

A suitable representation of the information we need to communicate makes the communication process easier and less error-prone. Also, we should be aware that what is considered a suitable representation may vary according to the situation: the stakeholder profile, the task under execution, the domain involved in the communication process. In this research, we focus on defining a suitable representation for DSML which are considered as important communication mechanisms in the Software Engineering and Information Systems domain. According to the notion that "a picture is worth than a thousand words", it is common that conceptual models are graphically represented using a Visual Modelling Language (VML). Currently the investigation of the development of a concrete syntax for VMLs has been limited. The main goal of this dissertation was the definition of an approach that improves the quality of the communication capabilities of VMLs through the establishement of a suitable design process. We claimed that this improvement in the quality of the communication could be achieved by combining mainly two well-known theories: The Unified Foundational Ontology (a foundational ontology) and the Physics of Notations (an information visualization theory), through the attachement of ontological guidelines and information visualization guidelines to the design process of the concrete syntax.

A survey that identifies publications, which describe the evaluation and the design of conceptual modeling concrete syntaxes, shows that only a few of them effectively use ontological theories. Some publications mentioned ontology, but they do not employ the ontology for analyzing or designing the concrete syntax of modeling languages. Nonetheless, they point out that the role of ontologies in the development of concrete syntaxes should be further investigated (FIGL; DERNTL, 2011)(MOODY, 2009). In summary, ontological theories have been recognized as an important mechanism to carry out evaluation and design of modeling languages, but the focus has been on abstract syntax, as in (AZEVEDO *et al.*, 2013). In the few cases that have investigated the relation between ontologies and concrete syntax, the focus has been on language analysis and redesigning an existing concrete syntax (GENON *et al.*, 2011) and not on design of new languages.

The relation between conceptual modeling languages and ontologies is already established. An ontology can be a conceptual model or can give rise to a language metamodel. It can be applied to evaluate a language and to (re)design it. In this research, we highlight another role in which an ontology can be applied: the design of a concrete syntax. Considering this possibility, we claim that ontological theories can be applied in the whole process of language design, not only as a final product (a conceptual model), but as part of the process (design and evaluation of abstract and concrete syntaxes). The ontology that is used throughout this dissertation was the Unified Foundational Ontology (UFO).

Based on UFO, a framework was developed that can be used for evaluation and definition of abstract syntax of modeling languages. The quality of this framework has already been successfully proved in a variety of works, conquering more and more adepts. A next step was visualized in (GUIZZARDI, 2013), in which Guizzardi exposed the idea that the UFO framework can be adapted to the design of the concrete syntax of VMLs, thus expanding the area of action of UFO and proposing that PoN and UFO can be put to work together. However, it was an initial step, since the ontological guidelines visualized by him are not part of a design process and a small set of OntoUML constructs were treated.

The information visualization theory was incorporated in this project by means of the Physics of Notations (PoN) theory. PoN is a set of principles whose quality and value for VMLs analysis and design are undeniable. However, it is still a theory subject to adaptations. Among other gaps, PoN is difficult to use in design tasks because it lacks a method. Different researchers and empirical studies pointed out this difficulty. In addition, PoN does not provide sufficient details on the application of its principles. Also, it indicates that there should exist a clear connection between the abstract and concrete syntax element without elaborating on this.

The research presented here has a strong theoretical foundation. Also, it reinforces the usefulness of empirical studies as part of the research process. Empirical studies were carried out from before the first version of the approach until a final test with the version presented in the text. Exploratory empirical studies and evaluation empirical studies were performed. Another aspect that stands out is the practical aspect of the approach application, presenting instructions for the probable users of PoN-S / PoNTO-S, as well as the elaboration of a text that allows a reader to acquire the knowledge necessary to apply the approach without reccuring to other information sources. Obviously, using other information sources could bring benefits as an in-depth awareness of his/her design rationale.

According to the research strategy adopted in this study and the results obtained from its execution, we consider that the goals established at the beginning of the work were met. The research goals are fulfilled in the form of the contributions described next.

6.2 Research Contributions

The main contribution of this work is the definition of a design process for the creation of DSML concrete syntaxes that can be applied both in DSMLs where the metamodel is based on ontologies (PoNTO-S) and in languages where the metamodel does not have such foundation (PoN-S). This is in accordance to the general research objective identified in Section 1.3.

Figure 42 identifies in green color the theoretical basis adopted in the research. In blue color we can notice the contributions of the research. PoN-S meets the first established specific objective for the research (*To define an approach for designing concrete syntax of DSMLs*[47] *applying Information Visualization theories in the format of information visualization guidelines*). PoNTO-S and the extension of the ontological guidelines are answers to the second specific objective (*To extend the approach previously defined to consider ontological theories, introducing ontological guidelines*). Finally, the evaluation empirical studies performed served to the third

[47] DSVMLs are commonly identified as simply DSML (Domain-Specific Modeling Language). So, in this work, we will apply the term DSML.

specific objective (*To apply the approaches developed in the design of DSML concrete syntaxes through empirical studies*).

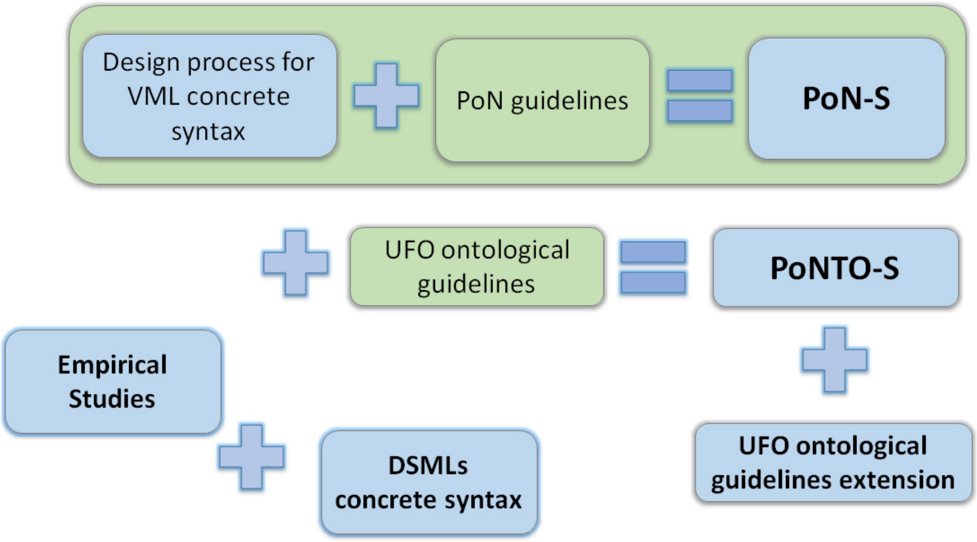


Figure 42: The research contributions

PoN, PoN-S and PoNTO-S are connected approaches – PoN-S is an expansion of PoN, and PoNTO-S is an expansion of PoN-S. Each one can be applied for designing of language concrete syntax separately. PoN and PoN-S do not incorporate ontological guidelines. On the other hand, the baseline of ontological guidelines described in (GUIZZARDI, 2013) and PoNTO-S are grounded on ontological considerations. PoN and the baseline of ontological guidelines do not establish a process and they are more suitable to perform evaluation activities instead of design ones, which does not mean that they can not be applied in design activities if the designer is an experienced professional. Complementing these two approaches, PoN-S and PoNTO-S define a design process and they are recommended to be applied in design activities, mainly to novice designers. We judge PoNTO-S as the more complete approach among the cited one, as it considers PoN principles, ontological guidelines and a design process.

The majority of PoNTO-S guidelines are based on (GUIZZARDI, 2013), which offers a baseline connection between PoN and UFO, through several mappings between OntoUML constructs and visual variables (as the one defined in PoN). PoNTO-S reorganizes these mappings in such way that they have an easier format to be applied by designers. Also, we deepened some guidelines. Besides that, new guidelines were proposed.

Both PoN-S and PoNTO-S have been evaluated through empirical studies. Several improvements have been made already, making them more suitable to be applied. New additions are under study.

Next, we describe in more details each of the cited contributions.

Contribution 1 – PoN-S

We developed a design process based on PoN starting from some basic design questions and a grouping of the different PoN principles, called as information visualization guidelines. The process begins with the specification of the dialect set using the VML abstract syntax as input. It ends with the validation of the proposed dialects. This process contains typical language design activities, as well as the control flow between these activities. For every activity, we also identify the PoN principles should be applied. We are supposing that PoN-S can be applied equally to DIML and DSML languages.

The proposed process values an in-depth analysis and understanding of the different artifacts used or generated throughout the process, establishing analysis and evaluation activities of these artifacts at various moments. The intention is that the designer should always have a deep understanding of the assets involved in the process. Also, the notion that the process occurs iteratively is contemplated, especially in the connection between the validation phase and the previous phases. In addition, several loops are presented in the process, indicating that some activities are performed in cycles.

To better guide the language developer, the process is graphically represented (through UML activity diagrams) and different abstraction levels are used. A description accompanies each diagram, explaining the main aspects of the diagram. In addition, there are instructions for every activity which describe the inputs, outputs, what to do and the connection with the PoN principles.

In addition to guide step by step the designer work, the process also suggests that the designer should produces supporting documentation for the language users, presenting the design rationale that s/he adopted in the design. This suggestion is in line with the idea that if users have an in-depth understanding of what they need to use, they probably perform their work better.

Contribution 2 – PoNTO-S

Where ontological theories fit into the proposed design process? They provide guidelines for the design activities which the PoN principles do not establish. In addition, since the ontological guidelines considered are those visualized by Guizzardi (2013), the connection between these guidelines and the design process also solved the UFO ontological guidelines lack that these guidelines did not have an application process in which to fit.

The ontological recommendations add a “how to do” aspect in the involved activity. To enable this, it turns mandatory that the metamodel is based on a reference ontology developed using OntoUML language. After a modeling element (sterotyped as an OntoUML construct) is selected and the designer should decide which representational element associate to it, s/he can do this being guided by the correspondent ontological guideline. Due to the detail level added by the ontological recommendations we believe that, in principle, they are more suitable to DSML languages.

Using ontological guidelines to establish the symbol set of a dialect reinforces the connection between the modeling elements and the correspondent representational elements. Due to this connection, a solid bound between the representational element and the correspondent real-world concept is also established. As a consequence, we expected that the modeling task quality applying the resulting DSML is increased because the visualization of a domain situation becomes closer to the reality.

Another benefit is that different DSML concrete syntaxes could be clearly connected if they adopt the same ontological guidelines. So, patterns are being established and this makes easier to design, learn and use different DSMLs.

This contribution is a clear advance in faciliting the communication process between stakeholders involved in a task using a model developed with a DSML considering the ontological guidelines, even if they don´t know the design rationale under theDSVML. They would know the represented domain.

Contribution 3 – PoNTO-S Ontological Extensions

After achieving some stability in binding the design process and the ontological guidelines, Guizzardi recommendations were restructured in a guideline format, deepened when it was necessary and new guidelines were added. Our main extensions are:

- We established an alternative representation of `roles`, when `relators` are represented, also representing `mediation` relations;
- When mapping `phase` construct, Guizzardi suggests applying color, brightness or texture. We identified when it is suitable to apply each of these visual variables and how to deal with `phase` hierarchies (see Appendix B);
- Guizzardi identifies that sometimes it is not necessary to represent abstract concepts. We detailed when representing or not representing abstract concepts (options to represent, or not, and how to drive it);
- We included a discussion if not representing `roles` or abstract concepts are cases of symbol deficit;
- We created an ontological guideline establishing a flexible sequence for choosing the modeling elements to be represented. This guideline is added in another design activity: Choose a model element to be represented;
- Also, there are sketchs of other ontological guidelines involving `modes`, `events`, `collectives`.

After these extensions, PoNTO-S is more complete and suitable for designing DSMLs whose reference ontologies have the most commonly used OntoUML constructs and relations.

Contribution 4 – The empirical studies

The empirical studies that we carried out also contribute to the research field. Especially the case studies involving the OPL language (that applied PoN-S) and LawV language (that applied PoNTO-S). We argue that the adoption of the design process improved the quality of these languages.

So, PoN-S and PoNTO-S are approaches that a language designer can apply to achieve a high-quality communication in the language concrete syntax. This is the manner in which our research influences other researches.

We recognize that the application of PoN-S and PoNTO-S demands a high designer capability. Even so, we claim that the proposed approaches can be helpful tools in the difficult task of producing a high quality VML.

The final version presented in this dissertation was the result of a long work, of theorical studies, of empirical evaluations and long discussion among the researchers.

While recognizing the degree of designer capability to properly apply the developed approach, we claim that PoN-S and PoNTO-S are a step further in creating a more complete and appropriate approach to VML design. However, there is a long way to go, as highlighted in the Section 6.4.

6.3 Which Approach to Apply, PoN-S or PoNTO-S?

Ontology-driven Conceptual Modeling is a relatively new field (VERDONCK *et al.*, 2015). As a consequence, the number of modeling languages currently in use that do not use ontologies is significantly higher than modeling languages based on ontologies. In order to be flexible and reach a larger number of modeling languages in development, we decided to isolate the ontological guidelines in the PONTO-S approach. In other words, we do not require from language designers that ontological guidelines should be applied in order for them to benefit from the approach. PoN-S is the design process without considering these ontological guidelines. Put differently, PoN-S does not require that the abstract syntax of the language under development is based on UFO.

Recognizing that a partial adoption of the proposed approach is possible does not reduce the importance of using ontological theories in the Conceptual Modeling field (GUIZZARDI, 2007). We highlight that the details offered by the ontological guidelines are important to perform the design activities (they add a “how to do” description to the affected activities).

Considering the successful use of UFO for (re)designing VMLs (GUIZZARDI *et al.,* 2015), we recommend applying PoNTO-S when developing new modeling languages, adopting OntoUML for creating the VML metamodel and considering the ontological guidelines for creating the corresponding VML concrete syntax.

6.4 Limitations and Future Research Directions

There has been a huge interest in VMLs, specifically DSMLs, and their application to improve productivity of different business fields. So, it is necessary to invest in methods and technologies to develop DSMLs.

PoNTO-S and PoN-S can be applied in their current releases. However, they can be improved. Their two main basis, PoN and UFO, are complexes basis. So, we decided to initiate with a smaller set of properties, the core ones. We plan to expand to new releases each time we achieve a balance in the presently stable release. Our future research directions are summarized in Figure 43. These directions are based on some gaps we already identified and listed as limitations of the research (see Section 20) and new plans visualized through the conduction of the research.

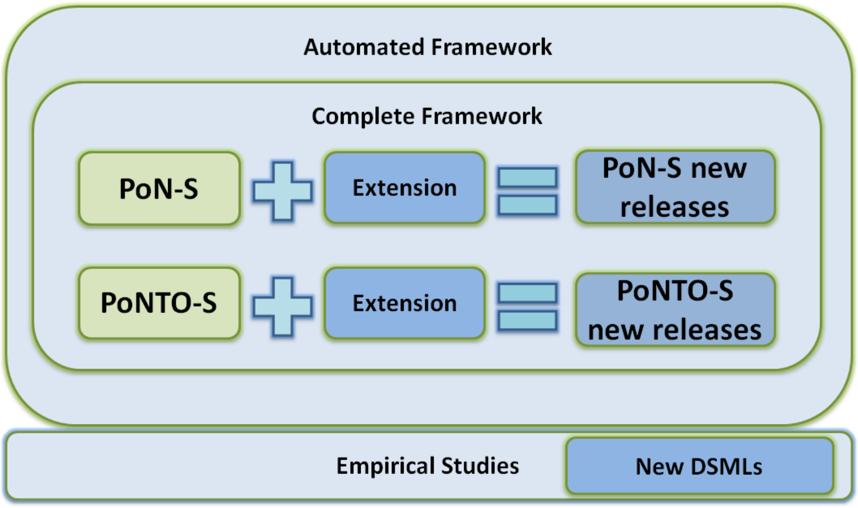


Figure 43: Future research directions

We already selected some specific characteristics we want to explore, as:

- During the development of PoN-S, we identified several activities, but only some activities were detailed. When presenting design questions, we identified three groups: dialect set, language level, sentence level. In the current release, we have concentrated pn the language level, although we have identified some activities related to the other levels. However, they were not sufficiently detailed here. Some examples include the following:

- Concerning the phase *Validate dialect(s)* and a possible evolution in the VML concrete syntax[48]: they can demand a return to previous phases. They are both difficult activities and demand updates in other phases to reflect their characteristics. So, dealing in detail with them demands a deepen study of their processes;
- Concerning the compound activity *Identify ways to manage model complexity*, we did not explore how complexity can be measured.
- We can define a more detailed guidance in how to specify the dialect set phase. For example, to better analyze how different stakeholder profiles can interfere in interpretation or development of models, using this information to define the characteristics of the dialect set;
- To study in which design activities other than designing the symbol set are possible to apply ontological guidelines and how to do this. As for example, in the *Identify ways to manage model complexity* activity maybe ontological patterns can be useful;

- We are dealing with a small set of OntoUML constructs. In a future moment, we would like to map other constructs to visual variables. Some examples are `collective`[49], `category`, `mixin`, `quantity`, `memberOf` relation, `subCollectiveOf` relation, `subQuantityOf` relation, `formal` relation. Also, we are just partially taking into consideration `substantial` types (only the set of sortal types), but almost no addressing at all `moment` types (except `relators`). Furthermore, we plan to include a more refined treatment of

[48] Evolution of the VML concrete syntax is not part of the design process. We are citing it here, because it demands a reapplication of the design process in a similar way it occurs during the Validation phase. Also, as a future perspective, we are planning to include the demands of the VML evolution, extending the design process.

[49] A draft for an ontological guideline to deal with `collective` construct and `memberOf` relation can be found in Appendix A.

taxonomies, for example, `relator-phases`, `relator-roles`[50], hierarchies involving `phases`, and so on. As OntoUML is not a complete mapping of UFO concepts yet, we intend to follow this adaptation as it occurs – when new constructs are added to OntoUML, we plan to deal with these new constructs in PoNTO-S;

- In its current release, PoNTO-S applies ontological guidelines in two design activities. We should study in which other design activities we can apply ontological guidelines and how to do this. As for example, in the *Identify ways to manage model complexity* activity maybe ontological patterns can be useful. The possibility of applying ontological pattern to reduce complexity of conceptual models is also discussed in (GUIZZARDI, 2014);
- To explore the ontological properties of relations and how we can use them in benefit of a better VML. For example:
  - We need to provide other alternatives for the representation of material relationships. In the current release, we offered two manners to represent this type of relationship. There are others, for example, the complete representation (`roles`, `relator`, `material` and `mediation` relations) and `roles` plus `material` relation without `relator` (SILVA TEIXEIRA *et al.*, 2014);
  - Representation of relation cardinality. There are relations in which the establishment of cardinalities is fundamental to a correct interpretation of the relation. For example, (i) to differentiate when a relation between two entity types is mandatory or optional; (ii) to avoid ambiguity when interpreting the relation, as discussed in (GUIZZARDI; WAGNER, 2008a); (iii) when the cardinality is specific (different from the traditional 0, 1 or N limits), e.g., to identify that a classroom can support at maximum 20 students. So, if the identification of inferior and superior multiplicity of relations is important to a suitable interpretation of a diagram, this diagram should have a suitable representation of these limits;

[50] This type of concept (`relator-phase, relator-role`) is under study and is not official yet.

- We intend to check if is it possible to use relations to define representation strategies to manage the complexity of models and how to do this;
- Current releases of PoN-S and PoNTO-S are focused on the primary notation of a VML. Representation strategies for managing model complexity and secondary notation issues did not receive a similar attention. An obvious curiosity is in what level these two issues can benefit from ontological properties;
- To build an automatized editor for developing DSMLs based on PoN-S and PoNTO-S. This editor could receive a reference model as input (probably an OntoUML model) and guide the designer during the design activies in an (semi)automatized manner. Possibly, adopting an inductive approach, as suggested in (GUIZZARDI *et al.*, 2011) for construction of conceptual models based on UFO.

In addition to include new features to PoN-S and PoNTO-S, we plan also perform new empirical studies, with a larger set of participants, with larger and more complex ontologies, with more comparison between different notations, benefiting the approaches with aspects not yet imagined. This new group of empirical studies should involve not only language designers as participants, but also language users (see Section 1.5 for more details about the type of empirical studies we intend to apply). In particular, empirical studies involving LawV are already being planned, as highlighted in section 5.2.2.

As a major ambition, in the future we could create a complete framework for the design and evolution of VMLs, contemplating from the creation and validation of the reference model - emphasizing the use of ontologies - through the creation and validation of the abstract and concrete syntaxes, as well as possessing support for the evolution of VMLs. This would be an automated and integrated environment, with support for different designer profiles. We could have integration with Menthor[51], with a symbol editor, or even with a database that would be constantly fed with typical symbols of different business areas.

[51] Menthor is an environment for development and validation of OntoUML models. For more details see (MENTHOR TEAM)

As described above, the development of this PhD research opens up a number of future work directions, being the origin of a long-term research project. PoN and UFO were the beginning. The current versions of PoN-S and PoNTO-S are expansions to such theories. In this section we identified our next plans to keep researching.

References

AG, S. ARIS - Business Process Analysis. At disposal in: <http://www2.softwareag.com/corporate/products/aris_alfabet/bpa/default.aspx>. Accessed in: 15/6/2017.

ALEXY, R. **A Theory of Constitutional Rights**. USA: Oxford University Press, 2009.

ARANDA, J.; ERNST, N.; HORKOFF, J.; EASTERBROOK, S. A Framework for Empirical Evaluation of Model Comprehensibility. Proceedings of the International Workshop on Modeling in Software Engineering. **Anais...** , MISE '07. p.7--, 2007.

AZEVEDO, C. L. B.; IACOB, M.-E.; ALMEIDA, J. P. A.; et al. An Ontology-Based Well-Founded Proposal for Modeling Resources and Capabilities in ArchiMate. In: D. Gasevic; M. Hatala; H. R. M. Nezhad; M. Reichert (Eds.); The Enterprise Computing Conference (EDOC). p.39–48, 2013.

BAAR, T. Correctly defined concrete syntax. **Software and Systems Modeling**, v. 7, n. 4, p. 383–398, 2008.

BERTIN, J. **Semiology of Graphics: Diagrams, Networks, Maps**. London, UK: University of Wisconsin Press, 1983.

BOBKOWSKA, A. Modeling Pragmatics for Visual Modeling Language Evaluation. Proceedings of the 4th International Workshop on Task Models and Diagrams. TAMODIA '05. p.75–78, 2005.

BOBKOWSKA, A. E. Visual dimensions of modeling languages in interdisciplinary perspective. 6th International Conference on Human System Interaction (HSI). 2013.

BOONE, S.; BERNAERT, M.; ROELENS, B.; MERTENS, S.; POELS, G. Evaluating and Improving the Visualisation of CHOOSE, an Enterprise Architecture Approach for SMEs. **The Practice of Enterprise Modeling**. p.87–102, 2014.

BRANDONE, A. C.; SALKIND, S. J.; GOLINKOFF, R. M.; HIRSH-PASEK, K. Language Development. In: G. G. Bear; K. M.) Minke (Eds.); **Children's needs III: Development, prevention, and intervention**. p.499–514, 2006.

BRUNSCHWIG, C. R. On Visual Law: Visual Legal Communication Practices and Their Scholarly Exploration. (E. S. et al. Zeichen und Zauber des Rechts: Festschrift für Friedrich Lachmayer, Ed.)**Editions WebLaw**, p. 899–933, 2014.

CAIRE, P.; GENON, N.; HEYMANS, P.; MOODY, D. L. Visual notation design 2.0: Towards user comprehensible requirements engineering notations. Requirements Engineering Conference (RE). p.115–124, 2013.

CHO, H.; GRAY, J.; SYRIANI, E. Creating visual Domain-Specific Modeling Languages from end-user demonstration. ICSE Workshop on Modeling in Software Engineering (MISE). 2012.

COSTAL, D.; GÓMEZ, C.; GUIZZARDI, G. Formal Semantics and Ontological Analysis for Understanding Subsetting, Specialization and Redefinition of Associations in {UML}. In: M. A. Jeusfeld; L. M. L. Delcambre; T. W. Ling (Eds.); Conceptual Modeling - {ER} 2011, 30th International Conference, {ER} 2011, Brussels, Belgium, October 31 - November 3, 2011. Lecture Notes in Computer Science. v. 6998, p.189–203, 2011.

ELAASAR, M.; LABICHE, Y. Diagram Definition: A Case Study with the UML Class Diagram. In: J. Whittle; T. Clark; T. Kuhne (Eds.); International Conference on Model Driven Engineering Languages and Systems (MoDELS). Lecture Notes in Computer Science. v. 6981, p.364–378, 2011.

EMMSAD'16. At disposal in: <http://www.emmsad.org/archive/2016>. Accessed in: 21/8/2017.

ERICKSON, J.; SIAU, K. Theoretical and Practical Complexity of Modeling Methods. **Commun. ACM**, v. 50, n. 8, p. 46–51, 2007.

EVERMANN, J.; WAND, Y. Towards Ontologically Based Semantics for UML Constructs. 20th International Conference on Conceptual Modeling - ER2001. **Anais...** . p.27–30, 2001.

FALBO, R. D. A.; GUIZZARDI, G.; GANGEMI, A.; PRESUTTI, V. Ontology Patterns: Clarifying Concepts and Terminology. Proceedings of the 4th Workshop on Ontology and Semantic Web

Patterns. 2013.

FALBO, R. DE A.; BARCELLOS, M. P.; NARDI, J. C.; GUIZZARDI, G. Organizing Ontology Design Patterns as Ontology Pattern Language. 10th Extended Semantic Web Conference - ESWC 2013-a. 2013.

FALBO, R. DE A.; QUIRINO, G. K.; NARDI, J. C.; et al. An Ontology Pattern Language for Service Modeling. Proceedings of the 31th Annual ACM Symposium on Applied Computing - ACM-SAC 2016. 2016.

FIGL, K. Symbol choice and memory of visual models. In: M. Erwig; G. Stapleton; G. Costagliola (Eds.); Symposium on Visual Languages and Human-centric Computing (VL/HCC). p.97–100, 2012.

FIGL, K.; DERNTL, M. The Impact of Perceived Cognitive Effectiveness on Perceived Usefulness of Visual Conceptual Modeling Languages. In: M. A. Jeusfeld; L. M. L. Delcambre; T. W. Ling (Eds.); International Conference on Conceptual Modeling (ER). Lecture Notes in Computer Science. v. 6998, p.78–91, 2011.

FISH, A.; STORRLE, H. Visual qualities of the Unified Modeling Language: Deficiencies and Improvements. Symposium on Visual Languages and Human-centric Computing (VL/HCC). p.41–49, 2007.

GEERTS, G. L.; MCCARTHY, W. E. An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. **International Journal of Accounting Information Systems**, v. 3, n. 1, p. 1–16, 2002.

GENON, N.; AMYOT, D.; HEYMANS, P. Analysing the cognitive effectiveness of the UCM visual notation. 6th International Conference on System Analysis and Modeling (SAM'10). 2010.

GENON, N.; CAIRE, P.; TOUSSAINT, H.; HEYMANS, P.; MOODY, D. L. Towards a More Semantically Transparent i* Visual Syntax. In: B. Regnell; D. E. Damian (Eds.); 18th international conference on Requirements Engineering: foundation for software quality (REFSQ). Lecture Notes in Computer Science. v. 7195, p.140–146, 2012.

GENON, N.; HEYMANS, P.; AMYOT, D. Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. Proceedings of the Third International Conference on Software Language Engineering. SLE'10. p.377–396, 2011.

GRIFFO, C. L.; ALMEIDA, J. P. A.; GUIZZARDI, G. Towards a Legal Core Ontology Based on Alexy´s Theory of Fundamental Rights. ICAIL Multi-Lingual Workshop on AI & Law (MWAIL 2015), 15th Conference on Artificial Intelligence and Law (ICAL 2015). 2015.

GRIFFO, C. L.; ALMEIDA, J. P. A.; GUIZZARDI, G. Legal Relations in a Core Ontology of Legal Aspects Based on Alexy´s Theory of Constitutional Rights. JURIX'16 Conference. 2016.

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. **Int. J. Hum.-Comput. Stud.**, v. 43, n. 5–6, p. 907–928, 1995.

GUARINO, N. Formal Ontology in Information Systems. Formal Ontology in Information Systems (FOIS). p.3–15, 1998.

GUARINO, N.; GUIZZARDI, G. We Need to Discuss the Relationship: Revisiting Relationships as Modeling Constructs. In: J. Zdravkovic; M. Kirikova; P. Johannesson (Eds.); **Advanced Information Systems Engineering: 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015**. p.279–294, 2015.

GUARINO, N.; GUIZZARDI, G. Relationships and Events: Towards a General Theory of Reification and Truthmaking. In: G. Adorni; S. Cagnoni; M. Gori; M. Maratea (Eds.); AI*IA 2016: Advances in Artificial Intelligence - XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 - December 1, 2016, Lecture Notes in Computer Science. v. 10037, p.237–249, 2016.

GUARINO, N.; OBERLE, D.; STAAB, S. What is an Ontology? In: S. Staab; R. Studer (Eds.); **Handbook on Ontologies**. Second ed., p.1–20, 2009.

GUIZZARDI, G.; FERREIRA PIRES, L.; VAN SINDEREN, M. On the role of Domain Ontologies in the Design of Domain-Specific Visual Languages. 2nd Workshop on Domain-Specific Visual Languages, 17th ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2002). 2002.

GUIZZARDI, G.; FERREIRA PIRES, L.; VAN SINDEREN, M. Ontology-Based Evaluation and Desing of Domain-Specific Visual Modeling Languages. 14th International Conference on Information Systems Development. 2005.

GUIZZARDI, G. **Ontological Foundations for Structural Conceptual Models**. 1st ed. The Netherlends: Centre for Telematics and Information Technology, University of Twente, 2005a.

GUIZZARDI, G. Agent Roles, Qua Individuals and the Counting Problem. In: A. F. Garcia; R. Choren; C. J. P. de Lucena; et al. (Eds.); Software Engineering for Multi-Agent Systems IV, Research Issues and Practical Applications [the book is a result of {SELMAS} 2005]. Lecture Notes in Computer Science. v. 3914, p.143–160, 2005b.

GUIZZARDI, G. On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. Proceedings of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference DB&IS'2006. p.18–39, 2007.

GUIZZARDI, G. The Problem of Transitivity of Part-Whole Relations in Conceptual Modeling Revisited. 21st International Conference on Advanced Information Systems Engineering (CAISE). 2009.

GUIZZARDI, G. Ontological Foundations for Conceptual Part-Whole Relations: The Case of Collectives and their Parts. 23rd International Conference on Advanced Information System Engineering (CAiSE'11). 2011.

GUIZZARDI, G. Ontology-Based Evaluation and Design of Visual Conceptual Modeling Languages. In: J. Reinhartz-Berger, I.; Sturm, A.; Clark, T.; Cohen, S.; Bettin (Ed.); **Domain Engineering. Product Lines, Languages and Conceptual Models**. 1st ed., p.345, 2013.

GUIZZARDI, G. Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling. In: E. Yu; G. Dobbie; M. Jarke; S. Purao (Eds.); **Conceptual Modeling: 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014**. p.13–27, 2014.

GUIZZARDI, G.; ALMEIDA FALBO, R. DE; GUIZZARDI, R. S. S. Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): the Case of the ODE Software Process Ontology. XI Iberoamerican Workshop on Requirements Engineering and Software Environments. 2008.

GUIZZARDI, G.; GRAÇAS, A. P. DAS; GUIZZARDI, R. Design Patterns and Inductive Modeling Rules to Support the Construction of Ontologically Well-Founded Conceptual Models in OntoUML. 3rd International Workshop on Ontology-Driven Information Systems (ODISE 2011). p.402–413, 2011.

GUIZZARDI, G.; HALPIN, T. Ontological Foundations for Conceptual Modelling. **Appl. Ontol.**, v. 3, n. 1–2, p. 1–12, 2008.

GUIZZARDI, G.; WAGNER, G. What's in a Relationship: An Ontological Analysis. 27th International Conference on Conceptual Modeling - ER2008. p.83–97, 2008a.

GUIZZARDI, G.; WAGNER, G. What's in a Relationship?: An Ontological Analysis. 27th International Conference on Conceptual Modeling (ER'2008). p.83–97, 2008b.

GUIZZARDI, G.; WAGNER, G.; ALMEIDA, J. P. A.; GUIZZARDI, R. S. S. Towards ontological foundations for conceptual modeling: The unified foundational ontology {(UFO)} story. **Applied Ontology**, v. 10, n. 3–4, p. 259–271, 2015.

GURR, C. A. Effective Diagrammatic Communication: Syntactic, Semantic and Pragmatic Issues. **Journal of Visual Languages and Computing**, v. 10, n. 4, p. 317–342, 1999.

HAAPIO, H.; PASSERA, S. Visual Law: What Lawyers Need to Learn from Information Designers. At disposal in: <https://blog.law.cornell.edu/voxpop/2013/05/15/visual-law-what-lawyers-need-to-learn-from-information-designers/>. Accessed in: 15/5/2017.

HENDERSON-SELLERS, B. Bridging Metamodels and Ontologies in Software Engineering. **J. Syst. Softw.**, v. 84, n. 2, p. 301–313, 2011.

HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. Design Science in Information Systems Research. **MIS Quarterly**, v. 28, n. 1, p. 75–105, 2004.

JURISTO, N.; MORENO, A. M. **Basics of Software Engineering Experimentation**. 1st ed. Springer Publishing Company, Incorporated, 2001.

KARAGIANNIS, D.; KUHN, H. Metamodelling Platforms. {EC-WEB} '02: Proceedings of the Third International Conference on E-Commerce and Web Technologies. 2002.

KARSAI, G.; KRAHN, H.; PINKERNELL, C.; et al. Design Guidelines for Domain Specific Languages. In: M. Rossi; J. Sprinkle; J. Gray; J.-P. Tolvanen (Eds.); Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM'09). p.7–13, 2009.

KROGSTIE, J. Evaluating UML Using a Generic Quality Framework. In: M. Khosrow-Pour (Ed.); **Encyclopedia of Information Science and Technology**. Second ed., p.6, 2009.

LANGER, P.; MAYERHOFER, T.; WIMMER, M.; KAPPEL, G. On the Usage of UML: Initial Results of Analyzing Open UML Models. In: H.-G. Fill; D. Karagiannis; U. Reimer (Eds.); Modellierung 2014, 19.-21. M{ä}rz 2014, Wien, {Ö}sterreich. LNI. v. 225, p.289–304, 2014.

VAN DER LINDEN, D.; ZAMANSKY, A.; HADAR, I. How Cognitively Effective is a Visual Notation? On the Inherent Difficulty of Operationalizing the Physics of Notations. 21st International Conference in Exploring Modelling Methods for Systems Analysis and Design (EMMSAD'16), Held at CAiSE 2016, Ljubljana, Slovenia. p.448–462, 2016.

MENTHOR TEAM. Menthor Editor. At disposal in: <http://www.menthor.net>. Accessed in: 30/5/2017.

MILLER, F. P.; VANDOME, A. F.; JOHN MCBREWSTER. **Mandamus**. USA: AlphaScript Publishing, 2010.

MISKE, C.; ROTHENBERGER, M. A.; PEFFERS, K. Towards a More Cognitively Effective Business Process Notation for Requirements Engineering. In: M. C. Tremblay; D. E. VanderMeer; M. A. Rothenberger; A. Gupta; V. Y. Yoon (Eds.); Advancing the Impact of Design Science: Moving from Theory to Practice - 9th International Conference, (DESRIST). Lecture Notes in Computer Science. v. 8463, p.360–367, 2014.

MOODY, D.; HILLEGERSBERG, J. VAN. Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams. In: D. Gaševic; R. Lämmel; E. Van Wyk (Eds.); **Software Language Engineering**, Lecture Notes in Computer Science. v. 5452, p.16–34, 2009.

MOODY, D. L. The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. **IEEE TSE**, v. 35(6), n. 756–779, p. 1–22, 2009.

MOODY, D. L.; HEYMANS, P.; MATULEVICIUS, R. Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. **Requirements Engineering**, v. 15, n. 2, p. 141–175, 2010.

MORRIS, S.; SPANOUDAKIS, G. UML: An Evaluation of the Visual Syntax of the Language. 34th Hawaii International Conference on System Sciences (HICSS). 2001.

MYLOPOULOS, J. Conceptual Modelling and Telos 1. In: P. Loucopoulos; R. Zicari (Eds.); **Conceptual Modeling, Databases, and Case: An Integrated View of Information Sytems Development**. 1st ed., p.49–68, 1992.

NELSON, H. J.; POELS, G.; GENERO, M.; PIATTINI, M. A conceptual modeling quality framework. **Software Quality Control**, v. 20, n. 1, p. 201–228, 2012.

OntoUML Model Repository. At disposal in: <http://www.menthor.net/model-repository.html>. Accessed in: 2/2/2017.

OPEN LAW LAB. Visual Law. At disposal in: <http://www.openlawlab.com/project-topics/illustrated-law-visualizations/>. Accessed in: 15/5/2017.

PARSONS, J.; COLE, L. What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. **Data & Knowledge Engineering**, v. 55, n. 3, p. 327–342, 2005.

PEIRCE, C. S.; BUCHLER, J. Logic as Semiotic: The Theory of Signs. **Philosophical Writings of Peirce**, Dover books on philosophy. p.98–119, 1940.

PETRE, M. Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. **Communications of the Association for Information Systems ACM**, v. 38, n. 6, p. 33–44, 1995.

RECKER, J. **Scientific Research in Information Systems: A Beginner's Guide**. Springer

Publishing Company, Incorporated, 2012.

ROEBUCK, K. **Model-driven architecture (MDA): High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors**. 1st ed. Emereo Publishing, 2011.

ROSA, M. LA; HOFSTEDE, A. H. M. TER; WOHED, P.; et al. Managing Process Model Complexity via Concrete Syntax Modifications. **IEEE Trans. Industrial Informatics**, v. 7, n. 2, p. 255–265, 2011.

SAGHAFI, A.; WAND, Y. Do Ontological Guidelines Improve Understandability of Conceptual Models? A Meta-analysis of Empirical Work. 47th Hawaii International Conference on System Sciences, {HICSS} 2014, Waikoloa, HI, USA, January 6-9, 2014. p.4609–4618, 2014.

SALES, T. P.; GUIZZARDI, G. Ontological anti-patterns: empirically uncovered error-prone structures in ontology-driven conceptual models. **Data Knowl. Eng.**, v. 99, p. 72–104, 2015.

SCHALLES, C. **Usability Evaluation of Modeling Languages - An Empirical Research Study**. Germany: Springer Gabler, 2013.

SCHERP, A.; SAATHOFF, C.; FRANZ, T.; STAAB, S. Designing core ontologies. **Applied Ontology**, v. 6, n. 3, p. 177–221, 2011.

SCHRAMM, W. How Communication Works. **The Process and Effects of Mass Communication**. p.3–26, 1955.

SIENA, A. **Engineering Law-Compliant Requirements: The N{ò}mos Framework**. 2010. University of Trento. 2010.

SILVA TEIXEIRA, M. DAS G. DA; FALBO, R. DE A.; GUIZZARDI, G. Can Ontologies Systematically Help in the Design of Domain-Specific Visual Languages? In: R. Meersman; H. Panetto; T. S. Dillon; et al. (Eds.); 12th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2013) - OTM Conferences. Lecture Notes in Computer Science. v. 8185, p.737–754, 2013.

SILVA TEIXEIRA, M. DAS G. DA; FALBO, R. DE A.; GUIZZARDI, G. Analyzing the Behavior of Modelers in Interpreting Relationships in Conceptual Models: An Empirical Study. 3rd International Workshop on Ontologies and Conceptual Modeling (Onto.Com 2014)/6th International Workshop on Ontology-Driven Information Systems (ODISE 2014). 2014.

SILVA TEIXEIRA, M. DAS G. DA; QUIRINO, G. K.; GAILLY, F.; et al. PoN-S: A Systematic Approach for Applying the Physics of Notation (PoN). In: R. Schmidt; W. Guédria; I. Bider; S. Guerreiro (Eds.); 21st International Conference in Exploring Modelling Methods for Systems Analysis and Design (EMMSAD'16), Held at CAiSE 2016, Ljubljana, Slovenia, Proceedings. p.432–447, 2016.

SONNENBERG, C.; HUEMER, C.; HOFREITER, B.; MAYRHOFER, D.; BRACCINI, A. The REA-DSL: A Domain Specific Modeling Language for Business Models. In: H. Mouratidis; C. Rolland (Eds.); **Advanced Information Systems Engineering: 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011**. p.252–266, 2011.

SOUSA, K.; VANDERDONCKT, J.; HENDERSON-SELLERS, B.; GONZALEZ-PEREZ, C. Evaluating a graphical notation for modelling software development methodologies. **Journal of Visual Languages & Computing**, v. 23, n. 4, p. 195–212, 2012.

STORRLE, H.; FISH, A. Towards an Operationalization of the "Physics of Notations" for the Analysis of Visual Languages. In: A. Moreira; B. Schatz; J. Gray; A. Vallecillo; P. J. Clarke (Eds.); International Conference on Model Driven Engineering Languages and Systems (MoDELS). Lecture Notes in Computer Science. v. 8107, p.104–120, 2013.

TOLVANEN, J.-P.; GRAY, J.; ROSSI, M. (EDS.). Domain-Specific Modeling with Visual Languages. **Journal of Visual Languages and Computing**, , n. Special Issue, 2004.

TVERSKY, B.; HEMENWAY, K. Objects, Parts and Categories. **Journal of Experimental Psychology**, v. 113, p. 169–193, 1984.

VERDONCK, M.; GAILLY, F.; CEZARE, S. DE; POELS, G. Ontology-driven conceptual modeling: a systematic literature mapping and review. **Applied Ontology**, v. 10, n. 3–4, p. 197–227, 2015.

WADE, A.; KOUTOUMANOU, E. Statistics Research & Methodology. At disposal in: <https://epilab.ich.ucl.ac.uk/coursematerial/statistics/index.html>. .

WAND, Y.; STOREY, V. C.; WEBER, R. An ontological analysis of the relationship construct in conceptual modeling. **ACM Trans. Database Syst.**, v. 24, n. 4, p. 494–528, 1999.

WAND, Y.; WEBER, R. Mario Bunge's ontology as a formal foundation for information systems concepts. In: E. D. Falkenberg; P. Lindgreen (Eds.); **Information system concepts: An in-depth analysis**. p.79–107, 1990.

WAND, Y.; WEBER, R. An Ontological Model of an Information System. **IEEE Trans. Softw. Eng.**, v. 16, n. 11, p. 1282–1292, 1990.

WARE, C. **Visual Thinking for Design**. Morgan Kaufmann, 2008.

WARE, C. **Information Visualisation - Perception for Design**. Third Edit ed. USA: Elsevier, 2013.

WEISSENBERGE, B.; VOGEL-HEUSER, B. Evaluation of a graphical modeling language for the specification of manufacturing execution systems. 17th International Conference on Emerging Technologies & Factory Automation (ETFA). p.1–7, 2012.

WOHLIN, C.; RUNESON, P.; HOST, M.; OHLSSON, M. C.; REGNELL, B. **Experimentation in Software Engineering**. Springer Science & Business Media, 2012.

WOUTERS, L. Towards the Notation-Driven Development of DSMLs. Model-Driven Engineering Languages and Systems - 16th International Conference, {MODELS} 2013, Miami, FL, USA, September 29 - October 4, 2013. Proceedings. p.522–537, 2013.

Appendix A. Draft of an Ontological Guideline to Deal with `Collective` Construct

As the case studies that applied PoNTO-S (Sections 5.2.1 and 5.2.2) do not include any `collective` type, we consider the treatment of `collective` construct as a draft and we exposed our proposal as an appendix section.

`Collective` represents collections whose parts play the same role with respect to the whole (complexes). For example, a forest (`collective`) is a collection of trees (`kind`). Collections bear a strong similarity to functional complexes (`kinds`). However, while in collectives all member parts play the same role (all trees are of the same type and are members of the forest), in functional complexes a variety of roles can be played by different components (GUIZZARDI, 2011).

When dealing with a collection, we should identify the members of this collection. In UFO this is done by the `memberOf` relation, a type of whole-part relation that connects the whole (`collective`) to its parts. `MemberOf` relation, unlike other whole-part relations, is intransitive (GUIZZARDI, 2011).

Update of the Ontological Guideline: Choosing a Model Element to Be Represented

To include the following item suggesting when to select collective constructs to be represented: As `collectives` are groups of entities instantiating different `sortal` types, they should be represented after defining the symbols representing the type of their members.

Update of the Ontological Guideline: Establishing a Symbol

Collective and MemberOf relation

As a `collective` represents a group of entities, it is natural to represent it as a group of the representational elements used to represent its members, that is, a symbol composed of various symbols representing the members grouped. Thus, to represent collectives, the following guideline applies: since collective is related to another entity representing its members, the symbol used to represent the members can be replicated to represent the collective, thus maintaining a visual connection between the collective and its members. So, the collective can be represented by a cluster of the same symbol used to represent the replicated members. In order to give an idea of group and at same time avoiding visual clutter, 3 or 4 member symbols are enough to represent a collective. Making use of two Gestalt laws - spatial proximity and similarity - it can be assumed that members are similar (in fact, they play the same role) and they should be represented in spatial proximity (WARE, 2013). To connect the collective to its members, the authors appeal to other Gestalt law: Connectedness (WARE, 2013). Thus, a solid line should be used to connect the members and the collective. Appealing to spatial grouping rather than spatial inclusion for representing this whole-part relationship also increases its discriminability to the other type of whole-part relationship (componentOf) addressed by PoNTO-S, to which we suggest using spatial inclusion for representing this relation type.

Figure 44 presents an example of the suggested representation. Figure 44(a) depicts the metamodel while Figure 44(b) depicts the application of the proposed visualization in an instance elements diagram.

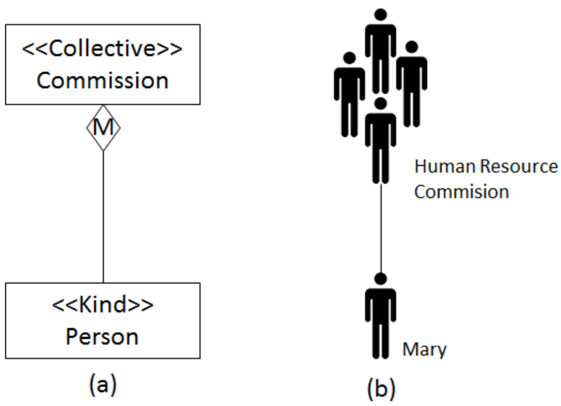


Figure 44: Example for representation of collective construct

We highlight that the text in Appendix B and Appendix C already consider the treatment of these updates.

Appendix B. Instructions to Language Designers Apply PoN-S and PoNTO-S

This appendix provides more detailed instructions for language designers performing PoNTO-S (or PoN-S) than those described in previous chapters.

To apply PoNTO-S, the designer should have prior knowledge of Modeling Language Engineering, UFO and PoN. To use PoN-S, the designer should have the same prior knowledge, except UFO. Even with such knowledge it is necessary to comprehend the PoNTO-S and PoN-S approachesp, so that the instructions described here can be executed in an appropriate manner.

For each activity of PoNTO-S / PoN-S approach we describe: (i) An identification (its name) – a clear meaning of the purpose of the activity; (ii) Its assets – main inputs and outputs; (iii) What to do in the activity – a brief description of the steps that should be accomplished in the activity; (iv) PoN principle(s) connection – identification of possible PoN principle(s) that should be taken into account in the activity execution; (v) How to do the activity – complementary guidelines of item (iv) responsible for enhancing the details of how to perform the activity - in this case through the application of ontological guidelines. The latter item is implemented only in case of PoNTO-S.

Once understood the selected approach, it can be executed through the design process described in Section 4.3. Following the process, for each non-compound activity identified in it, the detailed instructions are described here.

We recommend that in addition to providing the DSML for use, also information about each dialect designed should be at disposal in a complementary documentation, such as objectives, guidelines for use, guidelines followed during the design, that is, to present at least partially the design rationale adopted during language development. Knowing better how the dialect was created will help language users to understand it and probably to do a better using of the DSML.

Specify Dialect(s) Phase

Activities: Identify Dialect Requirements; Define Number of Dialects

These activities are described together due to their connection: to identify basic requirements of the language and based on them to define number of dialects that the language should possess to attend these requirements. The items described are:

- *Input*: VML abstract syntax – knowing the basis of the concrete syntax helps in understanding its requirements;
- *Output*: Modeling task, stakeholder profile, problem domain characteristics;
- *What to do*: Define the most common situation in which the language would be applied, that is, *Identify dialect requirements* – modeling task, stakeholder profile, problem domain characteristics. Using this information, decide if a concrete syntax should be organized in more than one dialect (*Define number of dialects*);
- *Related PoN principle*: Cognitive Fit. This principle advice to "use different visual dialects for different tasks and audiences" (MOODY, 2009), avoiding the common misinterpretation of "one size fits all". It should be a reminder to the designer that what is a good quality notation for one stakeholder profile or modeling task, maybe is not so good in a different situation. According to the Cognitive Fit Theory "problem solving performance is determined by a three-way fit between the problem representation, tasks characteristics and problem solver skills" (GENON *et al.*, 2010). Cognitive Fit principle seeks to establish visual flexibility for different stakeholder profiles considering that different designers could apply the language.

Activities: Identify Dialect Goal; Identify Dialect Directives

These activities are performed together due to their connection: to establish detail of each language dialect resulting in the so-called dialect set. The items described are:

- *Input*: dialect set requirements, VML abstract syntax;

- *Output*: VML dialect set – main characteristics that should be considered in the design of each dialect;
- *What to do*: After defining the main requirements of the language and the number of dialects it should possess, characterize each one of these dialects, establishing its goal and directives for design. In this activity, the designer should take into account besides the language abstract syntax, the influence relations (conflicts or synergies) that exist among PoN principles (see (MOODY, 2009)). It is not possible to establish the same level of compliance to all principles. So, the designer should choose the most important principles for each dialect in the language;
- *Related PoN principle*: Cognitive Fit. The same rationale described in the activities previously described is adopted here.

Implement Dialect(s) Phase

<u>Activities: Analyze VML Abstract Syntax; Analyze VML Dialect Set</u>

These activities are described together because they have the same goal: to obtain an in-depth comprehension of the inputs of the implementation phase. The items described are:

- *Input(s):* VML abstract syntax, VML dialect set – information necessary to generate each dialect;
- *Output(s):* There is no concrete output, but the expected result is an in-depth comprehension of the inputs;
- *What to do*: Perform a rigorous analysis of the language abstract syntax and the language dialect set in such a way that before implementing the language concrete syntax an in-depth comprehension of the language elements as well as language characteristic is achieved. This more in-depth comprehension should imply that when implementing each language construct or relation the designer is aware of the context in which this model element is included, making possible a better visual connection between representational elements. Also, the designer should be aware of what is expected in the

dialect considering that this information influences how the symbol set of each dialect should be implemented;

- *Related PoN principle*: Since this is a study activity, there is no direct connection to PoN principles, except for those possibly quoted in the dialect set.

Activity: Choose a Model Element to be Represented

The items described are:

- *Input: VML* abstract syntax;
- *Output:* A model element to be represented;
- *What to do*: Choose a model element (entity or relationship) to be sent to the next activity. Each model element should be chosen only once. Obviously, this is a loop situation – in each moment a different model element (without a corresponding representational element in the dialect) should be chosen, until every model element has been chosen. The designer should resort to the in-depth comprehension of abstract syntax and dialect set generated in previous activities as guidance. At the end of the loop execution, the designer should ensure that every model element was selected and that only elements identified in the language metamodel are represented;
- *Related PoN principle:* Semiotic Clarity. This activity is guided by the Semiotic Clarity principle to ensure that each model element will be represented by exactly one symbol, unless a different situation is required due to the directives established for the dialect. A 1:1 mapping between model elements and representational elements aids in the correct construction and visualization of diagrams. However, the symbol deficit anomaly seems to be a common situation on modeling languages (WEISSENBERGE; VOGEL-HEUSER, 2012)(GENON *et al.*, 2010), as an attempt to reduce language complexity;
- *How to do* (applying ontological guidelines – only in case of PoNTO-S): When choosing a model element to be represented, different considerations should

be done, as: (i) Is there an ideal sequence for choosing model elements? (ii) Should we represent abstract entities?

The designer should respect the following sequence when selecting a model element to be represented: in general, first represent entity types and then relations. An exception are `roles` that are entities types represented from relations. In representing entity types, start by representing rigid types (except `collectives`), then anti-rigid types (except `roles`), and finally `collective` types (even being a rigid type it should be represented after representation of the entity from which it originates, which, in turn, can be either of a rigid type or an anti-rigid type). In considering rigid types, address `kinds` and then `subkinds`. After representing rigid types, `phases` should be represented.

`Phases` always appear in a generalization set (GUIZZARDI, 2005a). Due to this reason, we suggest that representational elements of them should be defined together (all elements at same time), mapping the notion of generalization set from the metamodel to the concrete syntax. The joint definition of all `phases` in a generalization set is less error prone than if each `phase` in the generalization set is treated separately.

In cases of hierarchies, start at the top of the taxonomy and go down towards the bottom.

In case of multiple taxonomies starting from the same entity (for example, a `kind` possessing a `phase` partition and a `subKind` generalization set), the same rules above apply, that is, first treat the `subKind` generalization set and then the `phase` partition.

After representing entity types, we suggest the following order for the representation of relations: first represent `relators`[52], `roles` and `mediation` relations (it is worth remembering that the designer can choose not to represent such concepts, see section 5.1.2. In this case, the designer should move on to the next relation type). Next, whole-part relations should be

[52] `Relators` are not typical relations in UFO. `Relators` are `Moment Universals` (entities). However, given their conceptualization be strongly connected to events and relations and the way their representations were suggested, we chose to approximate them of relation representation.

represented (componentOf, memberOf). Finally, material relations are represented.

Concerning relator, roles and mediation relations. There are two alternatives: (i) To represent roles indirectly through the representation of the material relation between the substance sortals affected by the relator (which is also not represented); (ii) To represent roles, relator and mediation relations directly. The choice between these alternatives is a designer decision that should be in accordance to the language requirements. Furthermore, we suggest that the designer be consistent in his/her decision throughout the entire design process involving these elements.

Note: As a language increases in size and complexity, it is expected that the number and size of taxonomies also increase. In this case, more complex situations can arise, for example, a role specializing another role. In the alternative in which roles are not directly represented, how to deal with situations where we have roles from roles? That is, how do we visualize a connection between the involved roles? For example: the role Employee gives rise to role Manager (see Figure 36). How would not we lose this connection if we are not representing roles directly? In this case, we strongly recommend using a more complete representation (those showing relator and roles)[53]. Even more, this implies in a taxonomy of roles. So, this taxonomy would determine the order in which relators and the connected elements should be represented.

When there is more than one model element type in the same level (that is, more than one kind, or more than one subkind, or more than one phase, and so on) concerning the items above, it is not relevant the order which they are represented in.

[53] Another possibility could be to represent a subtype relation between the material relations involved (for details see (COSTAL *et al.*, 2011)). In this case the design could keep the indirect representation of roles. However, PoNTO-S first release does not deal with these specialized relations, so we would not recommend them.

The suggested sequence is flexible. The designer could represent first all `kinds`, next all `subkinds`, and so on. Or s/he could select one `kind` and explores the entire taxonomy underneath it (which may contain both rigid and anti-rigid types). Then, the designer could apply the same process to another `kind`, until all `kinds` and their taxonomies have been represented. For relations, they may be represented after all entity types have been mapped or each relation can be represented after representing the entity types involved in it. A special case, in which representation of entity types and relations are so strongly connected that they should be represented together is the set of concepts composed by `relator`, `roles` and `mediation` relations.

When selecting an entity type (`kind`, `subkind`, `phase`) the designer should consider if it is a concrete or an abstract type. If it is a concrete type, s/he should establish a symbol to represent it. If it is an abstract type, s/he could decide not to assign a symbol to represent it – this choice depends on designer decision according to the language requirements.

The decision of representing or not abstract types may vary with each occurrence of abstract entity. Even more, different dialects could deal differently with the identified abstract entities. It depends on each occurrence and on the designer decision. However, for the sake of standardization of the visual notation, we suggest that the decision of associating or not a symbol to abstract entities should be common to all abstract types of the language in the same dialect.

<u>Activities: Associate a Symbol to Model Element; Relate New Symbol to Already Defined Symbols</u>

These activities are described together because they result in the establishment of a new symbol. The items described are:

- *Input:* A model element, language dialect set, symbol s*et al*ready defined;
- *Output:* A new representational element;
- *What to do*: Define a representational element for the model element previously selected. This definition should consider the connection to real-word (semantic meaning). Also, the designer should relate the chosen symbol

to other symbols already defined in the dialect, respecting similarities and differences between concepts;

- *Related PoN principles*: Semiotic Clarity, Semantic Transparency, Perceptual Discriminability. These three principles are applied together because they are strictly related, resulting in the establishment of a symbol avoiding symbol anomalies, as overload and redundancy (Semiotic Clarity). Also, this activity is guided by the Semantic Transparency principle in order to establish a clear meaning to the symbol. Besides, these activities aim at evaluating the visual distance between the new symbol and others already defined, in accordance to their similarities and differences (Perceptual Discriminability).

We assume these two activities as core activities in the design process. Transparency and discriminability criteria are cited more than once in publications concerning concrete syntax as important criteria to be adopted, as in PoN (MOODY, 2009) and (KARSAI *et al.*, 2009). These activities are the moment to assure that concrete syntax considers these criteria.

Ideally, concrete syntax should reflect the ontological properties that exist in the metamodel, that is, each symbol or group of symbols connected should be based on the nature of the represented information identified by the correspondent element(s) in the metamodel (GUIZZARDI, 2013).

- *How to do* (applying ontological guidelines – only in case of PoNTO-S)

1. Map the chosen model element (entity type or relation) to a representational element following the mapping between OntoUML elements and visual variables described in Table 28, which is based on the guidelines proposed by Guizzardi (2013) and expanded here;

Table 28: Mapping of model element to representational element. Expanded from (GUIZZARDI, 2013)

OntoUML Construct	Visual Variable	Guideline
Entity		
`Kind`	Shape	Each `(sub)Kind` should be represented by a unique shape (the corresponding symbol). When establishing a new symbol, try following the

subKind		metaphorical resemblance between the representational element and the model element it represents. As in the Theory of Signs (PEIRCE; BUCHLER, 1940), a sign (representational element) can be: (i) An icon – it has a physical resemblance to the model element – for which the mapping between model element and representational element is direct. For example, a picture of a church to represent a church; (ii) An index – it possesses evidence of the model element it represents – where the mapping is mnemonic. For example, a smoke image to represent fire; (iii) An abstract symbol – it has no resemblance with the model element – and the mapping should be learned. It is an abstract sign. For example, a rectangle to represent a city. In case of an abstract (sub)Kind that the designer decided to represent, the representation of the whole taxonomy can occur in different moments, because different elements are involved: first, the root concept should have a symbol associated to it. This is the basic symbol of the taxonomy. Next, for each specialized subkind, a new symbol should be associated to it, adding some detail to the basic symbol, generating composite symbols. In this way, the involved representational elements are closed connected. For example, in a complete generalization set composed of <u>Person</u> (kind), <u>Man</u> (subkind) and <u>Woman</u> (subkind), the basic stick man index can represent a <u>Person</u>, a stick man with hat can represent a <u>Man</u>, a stick man with skirt can represent a <u>Woman</u>. The same idea (adding detail to a basic symbol) can be applied when only concrete elements are involved. Notes: - In case of an element involved in a whole-part relation, acting as the whole element, we suggest representing this element using a closed geometric shape, so that it is easier to show the part element(s) inside the whole element, as suggested in the guideline for whole-part relations; - In case of representation of relators, it is recommended that the interior part of the symbol representing a (sub)Kind is not blank. Example: Figure 37 and Figure 38 contains examples of representation and not representation of abstract types. <u>Person</u> is an abstract kind. In column 2, there is a symbol (a triangle) to represent it. <u>Man</u> and <u>Woman</u> (subkinds) are represented adding detail to the triangle. A hat to the <u>Man</u> and a star to the <u>Woman</u>. On the other hand, in column 3, the representations of <u>Man</u> and <u>Woman</u> are not connected to the

		representation of Person. This happens because after establishing an index to Person, the participant of the experiment decided it was not necessary (that is, this is a case in which the participant chose not to represent the abstract kind).
Phase	Color Brightness Texture	The representation of this type (always in a group of phases) should be a variation of values respecting the main shape that is used to represent the core concept ((sub)Kind) and the semantic distance (how close/different are the involved constructs) existing between each phase in the generalization set. In this manner, the visual identity is kept, but the model reader can identify some change on it, characterizing the variation of phases (reflected in the visual distance between symbols). An implementation recommendation is the application of color values. Other visual variables, such as brightness and texture, are also good options. They can be applied in different situations, according to the characteristics of the intrinsic property that gives rise to the phase partition: (i) If the intrinsic property has ordered characteristic, as for example, age (intrinsic property) of a Person (kind) that gives rise to a phase partition composed of Child, Adult, and Ancient, we recommend using a visual variable possessing the same characteristic (that is, being ordered) – brightness in this case; (ii) If the intrinsic property is not ordered, as for example, Active or Inactive situation (phase partition) of an Organization (kind), we recommend using a visual variable possessing this same characteristic (that is, being not ordered) – color in this case. Texture is another possibility. A common decision is the use of high-saturation color aspect (brightness) to represent extreme situations exploring a metaphorical relation between “more brightness” and “more quantity of something” (for example, in an overloading situation). Note: In a dialect in which the designer decided to represent relators, it is recommended that the interior part of the symbol representing a (sub)Kind is filled with the aspect (color, brightness) suggested above, keeping its border in black color (as the borders of all the other elements that are not relators). This is a manner to reserve boundary to deal with relators. Examples: In Figure 37 and Figure 38, Normal Workload Commission and

		Overloaded Work Commission are phases. From column 2 to 5 there are representations for them using color variation.
Collective	Composite shape	Each collective should be represented through a composite symbol. The basic symbol should be the symbol that represents the entity-part. This basic symbol should be replicated (3 or 4 times) creating the notion of grouping of elements. The replicated symbols should be positioned close to each other. Example: See Figure 44.
Role	-	Alternative 1 Roles should be represented indirectly through the representation of the relation among the (sub)Kinds involved in it. That is, roles are not represented directly. They are identified through the representation of the relations in which they are involved and that give rise to them. Examples: In Figure 37 and Figure 38, in column 5, all occurrences of roles are not directly represented. Alternative 2 See relator guideline.
Relations		
Relator Role Mediation relation	Shape Size Color Texture	The representation of relator, roles that are connected to it and mediation relations that connect the relator and its roles are described together since they basically result in a composite symbol, complemented by the substance sortal(s) that give(s) rise to the roles (that is, the allowed types). Associate a symbol – shape - (preferably an abstract symbol (geometric figure)) to each relator of the metamodel. As a (sub)Kind can be represented by a shape too, avoid that these two constructs have the same representation. If there is more than one relator, apply the Dual Coding principle, associating a name to each relator symbol, reinforcing the differentiation between the relators. To increase visual differentiation between relator representation and (sub)Kind representation, use some visual aspect (color, brightness, texture) in the border of the relator shape, while in a sub(kind) the visual aspect should be applied in its interior part. Also, while the interior part of the shape of a (sub)Kind representation should be filled, the

		interior part of the shape of a relator should be blank. If there is more than one relator, apply a different figure to each one. If there is a taxonomy of relators, apply variation in a basic shape. For example, rectangles with round or straight corners. Once the relator is represented, roles can be represented. For each role, duplicate the substance sortal symbol that gives rise to it (allowed type), reducing the size of the new symbol. The visual aspect (color, brightness, texture) of each new symbol should be the same of the relator symbol. If there is more than one role based on the same substance sortal, apply the Dual Coding principle, naming each symbol of each role. Or use an adding of detail. After creating the relator and roles, the connections in which they are involved should be represented: (i) Concerning the connection between roles and substance sortal(s) that give rise to them (allowed types), apply the traditional UML representation for subtyping: a line connecting the involved elements (each role and the involved substance sortal), and a small triangle near the supertype element; (ii) Concerning the connection between roles and relator, use an attachment position to the symbol, that is, place each role on the border of the relator symbol, characterizing that a role is strongly connected to a relator. Example: see Figure 36.
componentOf (whole-part relation)	Position	For each involved entity (whole and part elements), the designer should previously identify a symbol to represent it. These symbols should be preserved in the relation representation. To represent a componentOf relation implies in representing its meta-properties: (i) Irreflexive; (ii) Asymmetric; (iii) Transitive; (iv) Shareability (which can be of types shareable or not shareable); (v) Fependence (which can be of types existential or generic). Properties (i), (ii) and (iii) can be represented through variation in spatial inclusion (position visual variable). Spatial inclusion is itself an irreflexive and asymmetric relation. Implementation recommendations for the other metaproperties are: (a) Concerning a single whole-part relation: According to dependence type: (i) If it is a generic dependence - a part

		element(s) should be represented inside the whole element (an element inside a region). The traditional spatial inclusion representation conveys the idea that a part element should be part of a set (the whole element), but it can be placed in another set; (ii) If it is an existential dependence - a part element(s) should be represented as part of the whole element, like pieces of a puzzle (partitions of a region). This puzzle representation conveys the idea of inseparability between part and whole. A modeler cannot delete one without affecting the other; (b) Concerning multiple whole-part relations: Transitivity: to represent transitivity metaproperty of some whole-part relations use successive spatial inclusion – one element inside another as many times as necessary. The most internal element is part of the next element, which is part of the next element, and so on. Shareability: to represent the non-shareability metaproperty of some whole-part relations, use non-overlapping regions as whole elements, in which each region (or partition) indicates a whole element; To represent shareability property of some whole-part relations, apply spatial inclusion that show overlapping regions. These overlapping can be partial (the whole figure) or not. We recommend a proper overlapping, in such a way that it can be distinguished from the transitivity representation when it is possible and in case of a part element composing more than one whole element, this shareable element representation could be clearly visualized as participating in more than one representation of whole elements. Examples: In Figure 37 and Figure 38, there are occurrences of whole-part relations. Most of them have been represented through spatial inclusion. However, there are two occurrences of hierarchical chart representation, not following the recommendation expressed here.
`MemberOf` (whole-part relation)	Shape (line) Position	Use a line (a shape) connecting the elements involved in the relation to represent it – the `collective` and the entity-part. This is the most common manner to represent relational properties. Example: Figure 44.
`Material` relation	Shape (line) Position Texture	Use a line (a shape) connecting the elements involved in the relation to represent this relation. This is the most common manner to represent relational properties. If there is more than one relation to be represented, some alternatives can

		be applied, as for example: (a) The designer can apply different texture values in the line to differentiate the relations. Some possibilities to do this are: to vary dashed or dotted aspects of the line; to vary thickness of the line; (b) The designer can apply shape detail in the line to differentiate between the relations. For example, adding ornaments in the line extremities, such as triangles or arrows. Some relations demand complementary information that the designer should deal with. Some common situations are: (a) When the relation involves some type of hierarchical situation, as in the superior-subordinate relation, we suggest the use of position to complement the line. An above relation in the plane is a common representation for hierarchical situations; (b) When there is a direction in the relation, as for example in the relation "a client buys a product", the relation is from client to product, we suggest using shape detail – as an arrow or triangle near the line – to indicate the relation direction. This format makes the relation representation clear. Examples: In Figure 37 and Figure 38, there is a material relation (reportsTo). Columns 4 and 5 show this relation represented as a combination of line (shape) and position, different from each other.
Mediation	-	See Relator guideline.

2. Complementing the mapping indicated above, the designer should analyze taxonomies and the possibility of using redundant coding. This guideline comprises three activities, to be applied in different moments:

   (a) Check if the model element is part of some taxonomy in the metamodel and assure that taxonomical properties are also mapped to the visual notation.

   Representational elements should be similar/dissimilar to each other in accordance to their corresponding meaning and taxonomical structure identified in the reference ontology. This implies that, in general, an isomorphism should be established between the structure of the model elements and the corresponding structure of representational elements. This claim is completely valid when there are only concrete types (sub(kind), collectives, phases, roles[54] (alternative 2)) involved. In case of a role taxonomy (alternative 1), its representation is indirect – there is no direct representation. In this case, a model reader should be aware that the material relation involves participants performing a role even if this role is not explicitly represented. In case of abstract types, the designer could decide not to represent them. So, there would be no real isomorphism, but even in this case the mapping between model and representational elements would be correct;

(b) Review implementation choices to clearly reflect groups of symbols, in cases of multiple specializations involving the same (sub)Kind / collective / phase / role.

There are situations in which the same (sub)Kind will be root of different generalization sets. That is, a (sub)Kind could be specialized in subkinds (represented through variation in shape values), phases (represented through variation in color/brightness values) and roles (represented as a duplication of their ancestor). Each of these specializations could happen more than once. The designer should verify the impact of these combined representational elements in diagrams in which an instance element could be of a specific (sub)Kind, in a specific phase, performing a specific role. Similar situation can occur involving collective elements;

(c) Decide if it is helpful to reinforce some representational element(s) through redundant coding.

Applying redundant coding can be interesting and useful, but it should be applied carefully. The designer should identify what criteria can be used for choosing the elements to be highlighted through redundancy. Often, this depends on what type of visual query s/he wants to support with the diagram. So, the designer should analyze the

[54] Note that the suggested manner to represent the role already guarantees similarity with its allowed type, since the same symbol used for the allowed type is used in miniature for the corresponding role, attached to the relator symbol.

ontology structure and context before deciding which are those criteria. In the text below there are some situations that could be interesting to highlight applying redundant coding and suggestions of how to do this.

Redundant coding can be implemented in symbol boundaries to highlight some property of a representational element or group of elements. This can be made by using brightness or texture visual variables in the boundaries. Use of brightness or texture in the boundaries are useful to:

- Reinforce the distinction among entity types possessing some conflicting property, usually involving situations of flexibility of concepts. For example, in a diagram containing two types of whole-part relations with shareability (more flexible) and non-shareability (less flexible) properties, a designer can use different textures in the boundaries of the whole elements to clearly differentiate these two types of properties – non-shareability situation can be represented through a solid line and shareability situation can be represented through a dashed line;
- Reinforce a property of a representational element or group of elements in a diagram. For example, to highlight all rigid types.

Guidelines described here are influenced by the Semiotic Clarity principle. According to this principle, a designer should avoid, or at least reduce, symbol anomalies – redundancy and overload - when representing a model element (MOODY, 2009). As an exception, depending on goal and directives established to the dialect, some anomaly can be accepted. In general, this is not recommended.

Activity: Improve Use of Visual Variables

The items described are:

- *Input*: VML dialect set, VML abstract syntax, VML concrete syntax, representation strategies for managing model complexity – The first two are auxiliary information, the others are the sets that can be updated;

- *Output*: (revision of) VML concrete syntax, (revision of) representation strategies for managing model complexity;
- *What to do*: Review symbols and strategies created in previous steps, analyzing a possible modification in their visual variables values. Seek to maximize expressiveness of the symbols. The designer can do this reviewing individually each symbol or the symbol set. Keep in mind that to create a symbol it is possible to apply a single visual variable value, a combination of visual variables values (called composing code) or different symbols can be combined giving rise to another symbol (called composite symbol). The need for combination/composition can arise because it can be difficult to represent a model element applying only one visual variable/symbol or to reinforce discriminability through application of redundancy;
- *Related PoN principle*: Visual Expressiveness. The current activity is about the mechanisms a designer should pay attention to when creating a symbol set (or representation strategy). Basically, these mechanisms are visual variables, their values and the way they can be implemented together. The application of this principle occurs indirectly during the execution of other activities (when applying visual variables) and at this moment, performing a review of visual variables choices.

  Notes:

  - Try to explore as much as possible the visual variable set, taking the existing amount of model elements into account. Usually, for each symbol, the designer should start with a unique visual variable option - each entity or relation is displayed through a different visual variable. For example: `kind` (shape), `phase` (brightness) in OntoUML. As the amount of entities / relations increases, work with composing code (using more than one visual variable to generate one symbol) and composite symbols (a symbol defined as a combination of other symbols), as recommended by (STORRLE; FISH, 2013);
  - Explore family of visual variables to keep connection between similar concepts. This situation occurs, usually, in a taxonomy of constructs. For

example, similar shapes to represent similar `kinds`, similar brightness to represent similar `phases`, as argued by Guizzardi (GUIZZARDI, 2013).

Activity: Simplify Symbol Set

The items described are:

- *Input*: VML dialect set, VML abstract syntax, VML concrete syntax, representation strategies for managing model complexity – The first two are auxiliary information, the others are the sets that can be updated;
- *Output*: (revision of) VML concrete syntax, (revision of) representation strategies for managing model complexity;
- *What to do*: Review symbols and strategies created in previous steps, analyzing a possible modification in their visual variables values. The designer can do this reviewing individually each symbol or the symbol set as a whole. Simplify the symbols as deemed necessary: (i) Avoid creating unnecessary symbols; (ii) Avoid symbol redundancy; (iii) Apply as less visual variables as possible to characterize a symbol;
- *Related PoN principle*: Graphic Economy. This principle can be interpreted not only as "less is good", but as "keep as simple as possible". It should be a reminder to the designer that a good quality notation is a notation that has only the necessary and sufficient symbols, neither more nor less. It is a difficult activity to determine what is necessary and sufficient in a notation and possibly only empirical studies involving language users can assure this.

Moody *et al.* (MOODY *et al.*, 2010) claim that there are three strategies to deal with graphic complexity, namely: (i) Reduction of semantic complexity (changing language abstract syntax and therefore concrete syntax); (ii) Introduce symbol deficit (thus, affecting Semiotic Clarity principle); and (iii) Increase visual expressiveness (invest in increasing human capacity to discriminate between symbols rather than increasing number of symbols). From these three suggestions, we notice that reducing graphic complexity (which is the goal of Graphic Economy principle), in general, affects other principles (Semiotic Clarity, Visual Expressiveness, Perceptual

Discriminability). If this situation is a good or bad possibility, it depends on the dialect goal.

Activity: Define Textual Complement

The items described are:

- *Input*: VML dialect set, VML abstract syntax, VML concrete syntax, representation strategies for managing model complexity – The first two are auxiliary information, the others are the sets that can be updated;
- *Output*: (revision of) VML concrete syntax, (revision of) representation strategies for managing model complexity;
- *What to do*: Review symbols and strategies created in previous steps, analyzing benefits of a possible redundancy through use of textual information. The designer should do this when deemed that text will increase semantic expressiveness of symbols. The designer can do this reviewing individually each symbol or the symbol set as a whole;
- *Related PoN principle*: Dual Coding is the principle that is connected to redundant coding, but it refers to use of text instead of graphic as a way of referring to redundancy on representation. It should be applied when stakeholder profile is of a textual person, or in a situation which there is no suitable graphical symbol to clearly distinguish a concept.

When evaluating the benefits in applying this principle, we should consider that the use of text can overload a diagram (WEISSENBERGE; VOGEL-HEUSER, 2012). Thus, we should keep in mind which type of text benefits interpretation of diagrams instead of hindering the activity.

Note: Textual differentiation is cognitively ineffective (MOODY, 2009). Symbols that differ only by text have visual distance equal zero, that is, they are visually identical. Text is not a visual variable. So, unless the stakeholder profile is of textual profile or there is no suitable graphical symbol to clearly identify a concept, we recommend avoiding application of Dual Coding principle.

Activity: Evaluate Complexity of Symbol Set

As Erickson and Siau claim (ERICKSON; SIAU, 2007), the modeling language complexity has significant practical implications. So, this should be considered when developing a language. The items described are:

- *Input:* VML abstract syntax, VML concrete syntax, VML dialect set – information necessary to generate each dialect;
- *Output:* There is no concrete output, but the expected result is a more in-depth comprehension of the concrete syntax complexity;
- *What to do*: Apply some method to identify complexity of the concrete syntax. Once the designer is aware of the concrete syntax complexity s/he can collect evidence of such complexity that should be applied in the next activities. We suggest application of the complexity metric presented by Schalles (2013). Another possibility is suggested in (SOUSA *et al.*, 2012), in which authors performed a complexity analysis of a graphical notation based on (WARE, 2013)[55];
- *Related PoN principle*: As it is a study activity, there is no direct connection to a PoN principle except those possibly quoted in the dialect set.

Activities: Manage Model Complexity; Integrate Information from Different Diagrams

These two activities can be applied together resulting in a single representation strategy that is in accordance with both aspects of complexity management (organization and integration of information). The items described are:

- *Input*: VML dialect set, VML abstract syntax, VML concrete syntax – The first two are auxiliary information, the other is the one that can be updated;
- *Output*: (revision of) VML concrete syntax, representation strategies for managing model complexity (as many as the designer deems necessary);
- *What to do*: Evaluate the need to create representation strategies for managing model complexity based on the complexity evaluation performed in

[55] The authors used a previous edition of this book, published in 2004.

a previous activity. It can be created as many strategies as deemed necessary. In case of existing more than one dialect, evaluate if these strategies should be applied in different dialects. Probably this will be the case. Consider that smaller adjustments in the symbol set can be necessary depending on the dialect - symbols can change, but their meaning or their application process cannot. Each strategy can demand creating new symbols. So, these activities can affect both language and sentence levels (to manage model complexity we need that some representational elements exist - language level -, and a process to guide how to use these elements the best way as possible - sentence level);

- *PoN principles connection*: Complexity Management and Cognitive Integration. These principles are complementary to each other – how to organize information of a model and how to keep connection and traceability of these information. This can turn an essential issue of a notation, as diagrams increase in size and complexity. So, the importance of these principles increases as the amount of information being represented increases (MOODY, 2009).

  Notes:

  - The most known mechanism to manage model complexity is modularization (MOODY, 2009);
  - Moody *et al.* (2010) suggest some ways to decompose information. They can involve diagrams and diagrams elements. These ways are: (i) Diagram in another diagram(s), creating a diagram hierarchy – one diagram can be exploded in one or more diagrams; (ii) Element in another element(s), creating an elements hierarchy – one element in a diagram can be exploded in one or more elements; (iii) Element in diagram(s) (common modularization in which an element in a diagram represents another diagram). In all these cases, it should be possible to establish different abstraction levels, as many as necessary;
  - When recognizing that different diagrams would probably be developed, we should be aware of the need to integrate information spread across these diagrams. Moody *et al.* (2010) point out to the *Cognitive Integration*

of Diagrams theory as a way to do this. According to this theory, for multiple diagrams become cognitively efficient, we should propose mechanisms for two types of integration: cognitive (in the mind of the model reader) and perceptive (visual clues).

Validate Dialect(s) Phase

In the previous phases, some type of evaluation was carried out, as in the activities *Apply support principles* and *Evaluate complexity of symbol set.* However, rather than ensuring the language quality, the intent of evaluation in such activities is to provide information for the subsequent activities. Thus, an in-depth evaluation is still necessary to ensure language quality, verifying that theories and guidelines adopted in the process were followed, as well as verifying usability of the proposed dialect(s). This is the goal of the current phase.

The evaluation in this phase is divided into two parts: (i) An evaluation performed by the language designer, which is a mandatory activity; (ii) An evaluation performed with the target audience, which is an optional, but desirable, activity. We suggest that the designer evaluation should be concluded (the dialect set is completely evaluated by the language designer) before the second evaluation activity can be started.

In the first activity (more connected to language development evaluation), the objective is to detect eventual non-conformities to PoN principles. In addition to non-conformities, the resulting evaluation report may indicate corrective actions to be taken. The objective of the second evaluation activity executed by the language designer when building and evaluating instances of diagrams is to ensure the satisfaction of criteria such as user satisfaction and visual perception. Finally, an empirical evaluation should be conducted to assess the usability of the resulting concrete syntax, as well as user satisfaction, among other quality criteria that can be selected. Unlike the previous evaluation activities, the latter activity is conducted by possible language users as opposed to the language designer. So, the second group of evaluation is more connect to language use evaluation.

Activity: Check PoN Principles' Conformance

The items described are:

- *Input*: VML dialect set, VML concrete syntax, representation strategies for managing model complexity – The first item is a guidance to plan the evaluation. The other two itens are the itens under evaluation;
- *Output*: The evaluation can be positive (dialect is approved), positive with restrinctions or negative. In the last two cases, the proposed language dialect is not approved. If some problem is identified then a report of non-conformities and possible corrective actions should be generated;
- *What to do*: Check if the dialect is in conformance with PoN principles. This activity aims to verify if PoN principles are respected in the new dialect in accordance to the dialect goal and directives. We suggest as evaluation method the adoption of analysis characteristics of the PoN theory identified in (MOODY, 2009) (MOODY *et al.*, 2010). The nine principles should be checked considering dialect requirements. If the dialect is not approved, the *Specify dialect(s)* phase should be repetead in order to provide a more suitable dialect. In this case, the output of this activity (the report) should be taken into consideration. If the dialect is approved, the activity is concluded giving rise to the next activity;
- *PoN principles connection*: When performing *Check PoN principles' conformance* activity, we evaluate all PoN principles according to the dialect goal and directives.

Note: To facilitate this activity, we suggest that the language designer him/herself performs the activity. However, ideally, in activities like this, it is recommended that another professional conducts the evaluation of the PoN principles. This evaluation activity is correlated to Software Engineering quality assurance evaluation activities, in which there are specific roles (quality auditor) performing this type of activity. Thus, if another professional is available to perform this activity, this is the preferable situation.

Activities: Generate Diagram Instance; Evaluate Diagram Instance

These activities are described together because they result in an in-depth evaluation of each dialect by the designer. As in the *check PoN conformance activity*, these two activities result in an evaluation of the concrete syntax by the designer. The items described are:

- *Input*: VML dialect set, VML concrete syntax, representation strategies for managing model complexity, instance model(s) – The first item is a guide to plan the evaluation. The other two itens are the itens under evaluation. The next item is the input to some evaluation activities;
- *Output*: The evaluation can result positive (dialect is approved), positive with restrinctions or negative. In the last two cases, the dialect is not approved. If the dialect is not approved an evaluation report should be generated and the *Specify dialect(s)* phase should be repetead to solve eventual problems detected, thus, improving the resulting dialect. If the dialect is approved, the activity is concluded enabling the next activity;
- *What to do*: Evaluate instantiation(s). In (GUIZZARDI, 2013), the author suggests using visual simulation (analysis of model instances) for assuring that only intended state of affairs are represented by a model of the proposed language. In a similar manner, we suggest using visual instantiation to evaluate the concrete syntax of the designed language, checking if a diagram generated using the proposed concrete syntax is good enough (suggestion for judge what is "a good enough" concrete syntax is given in item (ii) below) for the intended model tasks ans stakeholder profiles. The activities are applied in sequence (*Generate diagram instances* then *Evaluate diagram instances*). In case of working with more than one diagram instance, the designer can decide how the loop will occur: for each instance, one should generate the instance and then evaluate it, or generate the set of instances and then evaluate the entire set.

  (i) Generate Diagram Instances. This activity aims at obtaining a set of diagram instances (as many as the designer deems necessary) that will be the input for the next activity. These diagrams can be type elements or instance elements diagrams, depending on the requirements of the language. Ideally, before designing the language concrete syntax, we assume that the language abstract syntax was designed and evaluated. The evaluation of the abstract syntax could generate model instances (as in the simulation process suggested in (GUIZZARDI, 2013)), and these model instances can be used as input to the evaluation of the concrete syntax;

(ii) Evaluate Diagram Instance. At this point, the designer should analyze a diagram generated using the proposed dialect and decide if this diagram is good enough for the intended model tasks and stakeholder profiles. To identify what is a "good enough" concrete syntax, the designer should adopt a framework for evaluation of visual notation, focusing on evaluating criteria such as user satisfaction and visual perceptibility, as defined by (SCHALLES, 2013)[56]. If the results of this activity indicate some problems, then *Specify dialect(s)* phase should be performed again in order to solve them;

- *PoN principles connection*: the PoN principles that should be involved depend on the dialect goal and directives. Their choice depends on considerations made during the performance of these activities.

Activities: Apply Empirical Study

The items described are:

- *Input*: VML dialect set, VML concrete syntax, representation strategies for managing model complexity – The first item is a guide to plan the evaluation. The other two itens are the itens under evaluation;
- *Output*: The evaluation can result positive (dialect is approved), positive with restrinctions or negative. In the last two cases, the dialect is not approved. If the dialect is not approved an evaluation report should be generated and the *Specify dialect(s)* phase should be repetead to solve eventual problems

[56] Some quality properties, other than Schalles work, that can be considered at this point are briefly presented in Section 2.1.

detected, thus, improving the resulting dialect. If the dialect is approved, the activity is concluded enabling the next activity;

- *What to do*: Evaluate the set of symbol and representation strategies of each dialect defined in the design process. The evaluation is performed by the target audience, an optional but desirable activity. At this moment, the designer should plan, perform and analyze the results of empirical studies (as many as the designer deems necessary). The study object of these empirical studies is the proposed dialect set, and the subjects should be the language target audience – participants possessing the stakeholder profile identified for the language. Considering that the use of VMLs involves two types of tasks, writing and reading (development and interpretation), we suggest that the empirical evaluation activities should analyze these two applications somehow. This is in line with some guidelines to do experimental evaluation of diagrammatical modeling languages proposed in (PARSONS; COLE, 2005), where the authors claim that “options for evaluation include: assessing the ability of developers to construct models that capture requirements ("write" tasks), and assessing the ability of readers of models to extract information contained in them ("read" tasks)”. If results of this activity indicate some problems, then the *Specify dialect(s)* phase should be repeated in order to solve them;
- *PoN principles connection*: PoN principles should be involved depending on dialect goal and directives. They should be considered during the design of the empirical study(ies).

Appendix C. The Set of OntoUML Constructs Considered in PoNTO-S

After considering the variety of constructs composing OntoUML, we decided to select a restricted set of its constructs to develop the current release of PoNTO-S. The obvious question is: which constructs to consider? In order to obtain a useful set of constructs we opted for considering the constructs most commonly used when developing OntoUML models. So, a new question arises: Which are these constructs?

In our first attempt to identify an ideal set of OntoUML constructs to be considered in PoNTO-S, we simply appealed to (GUIZZARDI, 2013), as this is the basis of the approach. Guizzardi worked with the following constructs: `kind`, `subKind`, `phase`, `role` (indirectly, that is, what we named here representation alternative 1), `componentOf` relation, `material` relation. Guidelines related only to these constructs were included in a version we term the pre-release of PoNTO-S. However, in (GUIZZARDI, 2013) there is no indication that the set of OntoUML constructs worked in it are the core ones, considering that this work was not intended to answer the question "Which are the most common used OntoUML constructs?".

An empirical study that analyzed the first impressions of the PoNTO-S pre-release was applied (see section 5.2.1). The participants of the experiment evaluated that direct representation of `roles` could be necessary. So, we decided to include a guideline related to representation of `roles` in a new release of PoNTO-S.

In parallel to this rationale, Sales and Guizzardi (SALES; GUIZZARDI, 2015) collected an OntoUML benchmark to analyze several issues related to OntoUML models, in particular ontological anti-patterns. This benchmark is partially exposed in ("OntoUML Model Repository"). Sales and other researchers are developing several statistics concerning the benchmark[57] composed currently of 54 OntoUML models of

[57] These statistics are not published yet, but they can be obtained when analyzing the benchmark composed of the 54 models.

different sizes and complexities. A basic statistic is the amount of OntoUML constructs used in the models (see Table 29).

Table 29: Number of occurences of UFO concepts in an OntoUML model repository of 54 models

OntoUML Construct	Number of Occurrences
Entities	
Role	1655
Subkind	1169
Relator	1089
Kind	837
RoleMixin	278
Mode	234
Category	216
Phase	207
Unknow Class[58]	123
Collective	115
Mixin	54
Quantity	23
TOTAL	**6000**
Relations	
Mediation	1718
ComponentOf	406
Material	375
Formal	325
Derivation	185
MemberOf	168
Unknown Association[59]	150
Characterization	132
SubCollectionOf	31
SubQuantityOf	2
TOTAL	**3492**

Based on Table 29 and on information previously identified we have expanded the set of OntoUML constructs to be addressed in the PoNTO-S first release to the following: (i) kind, subkind, phase, role (indirect representation – alternative 1), componentOf relation, material relation. These constructs have been already addressed in our baseline reference work and, as we see below, they are constructs possessing a high-occurrence in models (except the phase construct. However, its representation allows for a complete representation of all anti-rigid sortals); (ii) role (alternative 2 - direct representation) – besides being identified

[58] In several models, some entities were not stereotyped with some OntoUML construct. These occurrences were classified as "Unknown class".

[59] In several models, some relations were not stereotypes with some OntoUML relation. These occurrences were classified as “Unknow association”.

in an empirical study as a necessary construct to represent, Table 29 shows that this is actually the construct with the highest number of occurrences in the OntoUML benchmark; (iii) `relator, mediation` relation - due to the high number of occurrences in Table 29 and their strong connection to `roles`; (iv) `collective, memberOf` relation - even if they appear in Table 29 with a low number of occurrences, inclusion of such constructs is an advance to define ontological guidelines related to all `sortals`[60]. In addition, in the baseline exposed in (GUIZZARDI, 2013), there is evidence of application of `collectives`, albeit an implicit. In the DSML case illustrated there, for simplicity reasons, the commission concept was stereotyped as `kind`, even though it was also considered to be a collective of commission Members; (v) `modes`, due to the demands of LawV language (see Section 5.2.2) Here, only the `quantity sortal` is not represented (in Table 29 it has the lowest number of occurrences compared to other `sortals`). Thus, the new set of ontological guidelines partially covers `moments, substantials` and `relations`, besides it includes most commonly used constructs in OntoUML models.

[60] According to Guizzardi, in Figure 8-3 of his work (GUIZZARDI, 2005a), subtypes of `sortal universals` are: `kind, quantity, collective, subkind, phase, role`.