Using Goal Modeling to Capture Competency Questions in Ontology-based Systems

Paulo C. Barbosa Fernandes, Renata S.S. Guizzardi and Giancarlo Guizzardi

Ontology and Conceptual Modeling Research Group (NEMO)
Federal University of Espírito Santo (UFES), Brazil

paulo.barbosa.fernandes@gmail.com, {rguizzardi, gguizzardi}@inf.ufes.br

Abstract. Most of the available ontology engineering methodologies presuppose the existence of a set of questions which provide the objective and scope of the ontology under development. However, these so-called competency questions are not always clear from start. In this sense, such competency questions are comparable to system requirements, elicited and modeled during the requirements engineering stage of a software development process. Having this in mind, this work proposes the use of the Tropos methodology to help the ontology engineer reason and model competency questions. For that, he should follow the traditional Tropos process, starting with the early requirements activity to analyze the objectives of an organization. This initial stage is followed by the late requirements activity, in which the competency question are captured and linked to the organization's goals, following the goal modeling process. In this way, we aim at providing a methodology which will consistently drive the ontology engineer in developing an ontology from scratch.

Categories and Subject Descriptors: Data semantics, Knowledge modeling, Ontologies

Keywords: competency question, goal modeling, i*/Tropos, ontology

1. INTRODUCTION

In recent years, we have been watching an explosion of works involving the use of Ontologies, mainly motivated by the need for semantic interoperability among domain models and applications; this is due to growing interest on the Semantic Web [Guizzardi et al. 2009]. Practical benefits of an ontology-based system are, for example: a) allowing developers to reuse knowledge instead of reusing software; b) enabling the developer to reuse and share application domain knowledge using a common vocabulary across heterogeneous software platforms and programming languages; and c) letting developers focus on domain structure, preventing them from being distracted by implementation details.

To provide the aforementioned benefits, Ontologies are commonly used in computer science either as a reference model to support semantic interoperability, or as an artifact that should be efficiently represented to support tractable automated reasoning. We argue that in either case, a consistent ontology engineering methodology must be applied to develop Ontologies in a systematic way.

In this context, it is important to highlight that Ontologies inherently have a social nature, i.e. they are typically regarded as an explicit representation of a shared conceptualization, i.e., a concrete artifact representing a model of consensus within a community and a universe of discourse. That is why an ontology development begins with the definition of a set of questions, named competency questions, defining its objective, scope and expressiveness requirements. In other words, these are the question that a particular community of users thinks the ontology under development should answer.

In a sense, we can draw a parallel between an ontology's *competency questions* and a system *requirements*. If we think carefully, both a competency question and a requirement identify a future characteristic of the ontology (the former) and the system to be (the latter). In software, this is then translated into functionality while in an ontology, this is materialized by the right set of concepts and relations.

It is known that the requirements engineering activity must be thoroughly and precisely executed to guarantee the success of a software development project, thus being critical to assure software quality. Boehm [Boehm 1987] affirms that “Finding and fixing a software problem after delivery is 100 times more expensive than finding and fixing it during the requirements and early design activities.” What Boehm means is that omissions, contradictions, ambiguity, not discovered during the initial requirements will propagate to the other software development activities. And this can later result in a system full of bugs or in the need for an expensive and time-consuming redesign.

If we go back to the comparison between competency questions and requirements, we can than conclude that the elicitation and modeling of competency questions should also undertake a thorough and precise process. Curiously, this is not the case so far. Most ontology engineering methodologies presuppose the existence of this set of questions and concentrate their efforts in defining the next stages of ontology development.

This work is an attempt to provide a solution to fill in the gap between the definition of the competency questions and the ontology modeling per se. For that, we rely on goal modeling.

Goal modeling has been successfully applied in requirements engineering in the past years [Guizzardi 2007] [Liaskos et al. 2010] to deliver software for organizations in general. One of the main strengths of this approach comes from the importance it gives to the understanding of the current state of the organization (early requirements) before a solution is actually conceived. The analysis of the goals of the actors composing such organization allows one to answer questions such as: "What are the strategic objectives of the organization?", "Who share these goals?", "What alternatives are considered for achieving the goals?" and "What are the plans and resources applied to achieve such goals?"

Instead of solely focusing on system requirements, our proposal encourages the use of *goals*, *plans* and *resources* to capture the competency questions that will contribute to achieving the organizational goals. From such competency questions, some initial concepts are drawn and, from then on, the ontology can go through iterative conceptual modeling stages.

In this paper, we apply Tropos [Bresciani et al. 2004] as the goal modeling methodology to enable the definition of competency questions. The remaining of this paper is organized as follows. Section 2 we discuss the benefits of using an ontology-based system, Section 3 draws a parallel between the definition of competence questions and the requirements engineering activity, presenting the motivation for our approach. Section 4 describes some existing ontology engineering methodologies, particularly analyzing their support for the definition of competence questions. Section 5 presents our proposal, showing how we use Tropos to support the elicitation and modeling of competency questions. In Section 6 we describe a case study conducted in a real health care setting, using the proposed approach. Section 7 concludes the paper.

2. MOTIVATION: WHY ONTOLOGY-BASED SYSTEMS

Ontologies provide a number of useful features and serve multiple purposes in intelligent systems, as well as in knowledge representation and knowledge engineering in general. Gasevic, Djuric and Devedzic [Gasevic et al. 2009] describe two of these features: *vocabulary* and *taxonomy*.

An ontology provides a *vocabulary* for presenting the *terms of a given domain* in the way these terms are understood and applied by a community. This vocabulary comprehends logical statements that describe what the terms are and how they can or cannot be related to each other. Additionally, an ontology aims at specifying terms with unambiguous meanings, i.e. having their semantics independent of a particular reader and context. Thus, to sum up, an ontology-based system provides a vocabulary and a machine-processable common understanding of the terms of a particular domain.

Another interesting feature is *Taxonomy*. Taxonomy is a hierarchical categorization or classification of entities within a domain. It may, for example, serve the purpose of clustering entities based on common ontological characteristics. An ontology is inherently composed of a taxonomy, which may be codified in a machine-readable and machine processable form.

The ontology's vocabulary and taxonomy together provide a conceptual framework for discussion, analysis, and information retrieval in a given domain [Gasevic et al. 2009] [Guizzardi et al. 2007]. Bringing this to the health care domain, for example, they can support professionals and patients to acquire the right piece of knowledge in face of a certain health care problem or situation.

Gasevic, Djuric and Devedzic [Gasevic et al. 2009] also present *knowledge sharing* and *knowledge reuse* as two of the main purposes of a domain ontology. Both sharing and reuse are possible because of the common vocabulary provided by the ontology. Even if you do not consider any automation, having a common understanding of domain semantics enables people to communicate and cooperate better [Guizzardi et al. 2009]. Moreover, software applications may also benefit from machine processable ontological models, which allow different applications to interoperate and work in a more personalized and intelligent way [Guarino 1998].

For achieving the objectives of an ontology-based system, it is however paramount that its supporting ontology is well-structured and developed in a coherent and consistent manner. Otherwise, the vocabulary may lead to misleading interpretations and the taxonomy may undermine information retrieval, thus disabling effective knowledge sharing and reuse. This can be rather disastrous. Once again, the need for a systematic methodology for ontology engineering becomes clear.

3. THE RELATION BETWEEN ONTOLOGY ENGINEERING AND REQUIREMENTS ENGINEERING

Gruninger and Fox [Grüninger and Fox 1995] claim that defining competency questions is a way to determine the scope of the ontology and outline a list of questions that the ontology will be able to respond. Similarly, requirements engineering aims at the definition of the behavior of the software system, considering functional and non-functional requirements.

Besides the already cited Boehm, other researchers have commented on the importance of these initial stages of software development. Lutz [Lutz 1993], for instance, found that the inadequate definition of requirements is responsible for a significant portion of errors detected during the process of systems development, especially in the case of dedicated systems, real-time and critics. And Davis [Davis 1993] claims that eliminating engineering errors becomes increasingly difficult and expensive as the system progresses to later stages of their life cycle, such as the implementation stage.

To the extent of our knowledge, there is no study on the impact of a poor definition of the competency questions on the success of the ontology. However, we would expect it to be no different than what the requirement engineering researchers have found.

Imagine, for instance that an ontology-based system has been developed for a travel agency. After the system is already running, the owner tries to run the following query: "What are the touristic attractions of Rio de Janeiro this week"? However, suppose that the only competency question regarding touristic attractions, defined before modeling the ontology was "What are the touristic attractions of a particular city?" Since such question did not take any temporal aspect into account, the travel agency owner's query cannot obtain the answer to his query.

This exemplifies how an incomplete competency questions definition can undermine the success of an ontology, as the basis of a software system. More complex examples could be conceived, including examples in which such information loss could seriously harm system interoperability or even blur the communication within organizations (without considering any automation whatsoever).

Although the importance of this definition step, within the field of ontology engineering, the existing methodologies make little use of elicitation and modeling techniques. And moreover, these methodologies still do not satisfactorily consider guidelines for the identification of concepts and relations [Campos 2010]. Consequently, the ontology engineer remains so far with very limited resources for requirements for ontology engineering in the process of ontology development.

4. COMPETENCY QUESTIONS DEFINITION SUPPORT IN EXISTING ONTOLOGY ENGINEERING METHODOLOGIES

Soares [Soares 2009] present an analytical study includes a list of 30 methodologies to build ontologies. For example study selected in this work eight well-known methodologies for building ontologies.

To understand how each method deals with requirements engineering, some of the results were extracted and shown in Table I, which is then followed by a brief description of each methodology.

Table I. Methodology vs. Technique for the definition of competency questions

Methodology	CQ Definition Technique
Sensus	Absent
Cyc	Manual extraction of the knowledge required
Grüninger and Fox	Informal Competency Questions
Uschold and King	Determine the purpose of the ontology
Kactus	List of requirements
Methontology	Defining the scope of the ontology
Method 101	Informal Competency Questions
Neon Methodology	Competency Questions in natural language

The Sensus methodology has no technique for determining competency questions. Following the methodology steps, in order to define the requirements, one should identify key terms of the domain and manually link these terms to the SENSUS ontology.

Cyc is a large common sense knowledge base, in which the first step to extract common sense knowledge is a manual extraction analyzing books, journals and articles, identifying issues that could be answered after reading these texts [Lenat et al. 1990] [Lenat 1995]. The reason why the Cyc knowledge base can be considered an ontology is because it can be useful as substrate for building different intelligent systems and also as a base for their communication and interoperability [Gómes-Pérez et al. 2004].

Michael Grüninger and Mark Fox in 1995 [Grüninger and Fox 1995] presented the methodology used for elaboration of the TOVE ontology. The first two steps were defined: developing motivation scenarios, which aim to identify problems in the current environment and specify informal competency question. These informal competency questions specify natural language requirements that the ontology should be able to attend. The example of an informal competency question could be: "Considering a young traveler, with limited budget for accommodation, what kinds of accommodations would be available?" [Fox 1992].

Uschold and King (1995) consider as a first step to identify the purpose of the ontology under development, which aims at identifying: a) the need for development; b) the intended uses of the ontology; c) the relevant terms of the domain, For example, considering an ontology in the domain of interest of a travel agency, these terms could be: places, types of places, accommodation, types of accommodation, trains, buses, etc. [Uschold and King 1995] [Uschold et al. 1998].

Kactus [Bernaras et al. 1996], the first step would be to develop a list of needs or requirements that must be met by the application. Again, using a travel agency as example, an example of requirement could be: "Get the most suitable place for the customer, etc." [Gómes-Pérez et al. 2004].

The Methontology [Gómes-Pérez et al. 2004] includes a set of development stages (specification, conceptualization, formalization, integration, implementation and maintenance), and techniques to accomplish the planning, development and support of ontologies. In the specification activity, the following items should be defined: the purpose of ontology (what is intended), its usage scenario, scope and who the end-users are. This can be done by generating a document in natural language [Guizzardi 2007].

The Method 101 proposes, as the first step, to determine the domain and scope of the ontology, answering questions such as: "What will we use the ontology for?", "What are the questions to which the ontology should provide answers?" [Noy and Guinness 2001].

The Neon methodology proposes the following tasks for ontology specification: Identify purpose, scope and level of formality, Identify intended users, Identify intended uses, Identify requirements, Group requirements, Validate the set of requirements, Prioritize requirements and Extract terminology and its frequency [Gómez-Pérez and Suárez-Figueroa 2009] [Suárez-Figueroa et al. 2008]. The goal of Identify requirements task is to obtain the set of requirements or needs that the ontology should fulfill. Users, domain experts and the ontology development team carry out this task taking as input a set of ontological needs for identifying the ontology requirements, using techniques as writing the requirements in natural language in the form of the competency questions. The output of this task is a list of competency questions written in Natural Language and a set of answers for the competency questions [Suárez-Figueroa et al. 2008].

5. GOAL-BASED MODELING METHOD TO CAPTURE AND MODEL COMPETENCY QUESTIONS

This section represents the core of this paper, presenting our contribution to support competency questions via goal modeling. In subsection 5.1, we justify the use of Tropos in our work; subsection 5.2 summarizes our proposed Ontology Engineering approach, setting the context for the work presented in this paper; and subsection 5.3 explains in detail how we apply Tropos to capture and model competency questions.

5.1 Applying Tropos

Analyzing the most common methods found in the literature for the development of ontologies and their approaches to specify competency questions (see Table I), we can identify three main objectives:

i. defining the scope of the ontology, i.e. what parts of a specific domain the ontology should cover;
ii. deciding the ontology's applicability or in other words, what the ontology will be actually used for;
iii. specifying what questions the ontology must answer, i.e. what inferences the ontology should allow.

At this point, we believe that we can successfully apply Tropos [Bresciani et al. 2004] to achieve these goals. The following paragraphs justify our choice.

Tropos is an agent-oriented software engineering methodology which is founded on the concepts of *actor* and *goal*. Tropos's early requirements' activity is concerned with understanding the organizational setting as it is, before a solution is actually conceived. The output of this activity is an organizational model, which includes relevant actors, their goals and interdependencies. By focusing on the problems which an organization aims at solving, the early requirements model provides us with the right context for the definition of the scope of the ontology (objective i). Moreover, by modeling actors, along with their goals and interdependencies, we gradually refine the model, up to a point in which the ontology's applicability becomes clear (objective ii)

In Tropos's late requirements activity, a solution is posed, generally in the form of a multiagent system. Still in this stage, the system is modeled in its operational environment by describing its most important functions. For that, the system under development is also represented by an actor who has a number of dependencies with other actors in the organization. These dependencies define the functional and nonfunctional requirements of the system. Such requirements are directly linked to the aforementioned objective iii, as it defines in detail what kind of support a system (in our case, an ontology-based system) should provide.

Finally, modeling *resources* provides us with a suitable concept for information modeling. Thus we can directly apply a Tropos concept to visually model competency questions and later, some initial concepts of the ontology.

To some up, the systematic analysis of goals provided by Tropos gradually leads us to understand what are the objectives involved in an organizational environment. And from this, we can understand that to satisfy these goals we need the right kind of information, so we can define the competency question based on this information need.

5.2 Proposed Approach

Fig. 1 depicts the proposed Ontology Engineering approach. In *early requirements*, the goals of actor **A1** of a given organization are analyzed, following the analysis methods advocated by Tropos. In *late requirements*, during this activity, competency questions are defined to accomplish the goals of this actor, when a solution is finally conceived in the form of an ontology-based system, dependencies are established between the system actor **A2** and actor **A1,** the former acquiring competency questions **CP1** and **CP2** from the latter.

Besides modeling the competency questions, it is useful to elicit a few concepts, which directly derive from such questions. These concepts will later compose the actual ontology model. So still in late requirements, initial concepts of the ontology (here illustrated by **c1**, in diagram of actor **A2**) are captured and linked to the competency questions. In the ontology modeling activity, such initial concepts are specified and refined, applying a different conceptual modeling language, such as UML or ER diagram, for example.

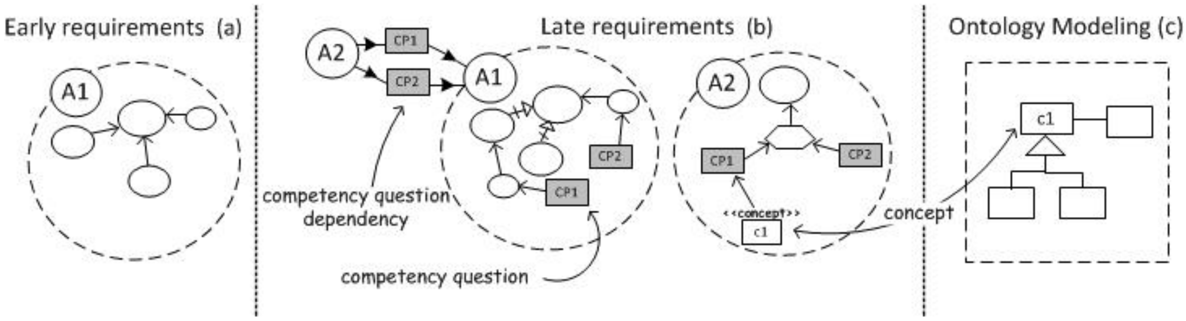


Fig. 1. Proposed Ontology Engineering Approach

The ontology created in the ontology modeling activity, can be used both as a conceptual reference model (design time), as a model to be used computationally (runtime) allowing, for example, the use of an inference engine. Furthermore, the ontology (design time) could also be translated into another model, for instance, the model database.

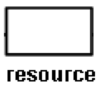
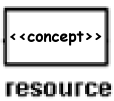
It is important to state that in the ontology modeling activity, we apply the UFO foundational ontology [Guizzardi 2005] to guide us on the ontology development process. Foundational ontologies have been recently recognized as important tools to guarantee semantic coherence and consistency when developing domain ontologies.

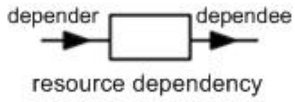
The difference between a foundational ontology and a domain ontology has to do with purpose and focus. While the latter is aimed at capturing a particular domain of discourse, the former is supposed to be domain-independent, dealing with general definitions such as *entity, part-whole relation*, and so on. In general, applying UFO provides us with safe guidance when deciding, for instance, if a concept should be seen as an entity or a property, also assisting us in finding out how two entities should be properly related. A deeper discussion on this topic, along with a detailed description of the UFO ontology is out of the scope of this paper and can be found in earlier publications such as [Guizzardi 2005] [Guizzardi et al. 2009].

5.3 The applied Tropos constructs to capture competence questions

To model the information requirements, which in our case, are the competency questions, Tropos uses the constructs presented in Table II. The second column of this table specifies how we apply such constructs.

Table II. Tropos Constructs to capture the competency questions

	Tropos element	**Proposed use**
A		Used to identify the competency questions.
B		Extension of the resource construct to model the concepts involved in competency question. These concepts are the initial concepts composing the ontology.

C		Used to link competency questions to goals and concepts to competency questions.
D		Used to represent the dependence of the competence question.

The Tropos's resource construct (row A on Table II) is naturally used to identify competency questions, since they are described as representing "a physical or an informational entity" ([Bresciani et al. 2004], p.207). A competency question is clearly an informational entity and in our case, such entity is a resource to achieve the organizational actors' goals and later the ontology-based system goals. To model that, we connect these resources to goals via a *means-end link* (column C on Table II). This means that the necessary information for the goal to be achieved is provided by the resource that represents the competency question. Depending on the complexity of the goal, including its information needs, several competency questions may be linked to the same goal.

As stated in subsection 5.2, we propose some of the initial concepts composing the ontology are captured during the late requirements activity. To model such concepts in Tropos, we propose an extension, annotating these special kinds of resources as <<concept>>, applying UML stereotypes (row B on Table II). Please note that this element does not provide the regular semantics of a resource, since it would be unnatural be consider them as resources to achieve goals or plans. This justifies our extension. Something else we must care for is how to relate these concepts with the other Tropos's constructs. In this case, concepts can be only connected to competency questions, also via means-end link (row C on table II). We find this a natural choice, since by the means of modeling specific concepts in the ontology; we are able to infer knowledge, thus answering a specific competency question.

A dependency relation (row D on Table II) is applied to represent that an actor **A2** depends on an actor **A1** to acquire a given competency question which will later guide the ontology engineer in the ontology modeling activity. In this case, the dependence is a resource representing such competency question.

In terms of the process, there is an important observation to be made. In some cases, we know in advance that the system under development will be based on ontologies, even before starting the Tropos modeling activity. In this case, the competency questions are raised still in the perspective of the organization's actors. And later, such questions are delegated to the system actor. However, we also consider the case in which only in late requirements, a decision is made to deliver an ontology-based system. In this case, perhaps many of the system's goals, tasks and resources have already been elicited. However, it is important that the early requirements model is then revisited and competency questions modeling are normally performed.

6. CASE STUDY

This section aims at illustrating our approach, by describing part of a case study conducted in a basic health unit in the city of Vitoria, Espirito Santo, Brazil. Please note that we do not mean to be exhaustive and many interesting information regarding the study is out of the scope. Our main objective here is to illustrate the approach described in section 5.

Subsection 6.1 introduces some concepts regarding the Brazilian primary health care; in subsection 6.2, we show an early requirements model, focusing on the goals of a *community health agent*; subsection 6.3 focuses on the late requirements model, when the competency questions are modeled and delegated to the system-to-be; and finally, in subsection 6.4, we show an ontology excerpt to illustrate how we go from the Tropos models to a UML conceptual model, guided by UFO.

6.1 Case Study Context: The Brazilian Family Health Program

With the objective of reorganizing the primary health care practices on new basis and criteria, the Brazilian Ministry of Health designed the Family Health Program (FHP) in 1994 [Franco and Merhy 1999]. This program is characterized by having the family as the unit of action instead of the individual. Additionally, FHP seeks for the involvement of these families and the community as a whole to solve health related problems.

Epidemiology [Filho and Rouquayrol 2002] plays an important role to enable FHP. This medical field seeks to understand the factors that determine the frequency and distribution of diseases in humans. By basing the decision-making process on epidemiological studies, FHP introduces an active view on health care, in which health care institutions should preemptively act to avoid people from getting sick [Lefevre and Lefevre 2007].

6.2 Early Requirements: the Goals of the Community Health Agent

This section presents and analyzes an excerpt of the case study's early requirements model. A key role in the FHP is played by the community health agents, who are generally responsible for monitoring family health. Fig. 2[1] models this actor's main goals, plans and applied resources.

The Community Health Agent actor has a general softgoal of "*improve families health*", which leads to the main goal of "monitoring families' health and life conditions". This goal is then decomposed in three important sub-goals: "*registering families*" at the FHP if the patient is still not registered, "*keeping the family's information up to date*" and "*guiding families*" so that they can themselves act on behalf of their own health.

An important resource for accomplishing both the "*registering families*" and the "*keeping the family's information up to date*" goal is the "*form A*", which contains all the relevant information about the family. This form is filled in in the first visit of a family member and should always be kept up to date. Updating the form will be necessary, for example, if some of the information (e.g. the address of the family, the number of family members, etc.) was changed since the last visit of a family member.

The "*promoting education and vigilance actions*" plan contributes to "*guiding families*", as these families gradually learn to maintain pro-health habits and to play a more active role in keeping the authorities informed of any risk situation. The "*know the epidemiological profile of patients in a given region*" goal also greatly contributes to "*guiding families*". This means that the Community Health Agent will understand the conditions of the region where the family lives, preventing disease before it proliferates among the other families in the region.

The "*monitoring families*" goal positively contributes to "*mapping families at risk*". This is also a very important means for "*improve families health*". For instance, after mapping the regions more affected by the dengue fever, the government could more intelligently direct resources to these regions. This could have a very fast positive effect on the number of cases in such regions.

[1] The Tropos's models designed for this paper have been made with the support of TAOM4E, available at http://selab.fbk.eu/taom/

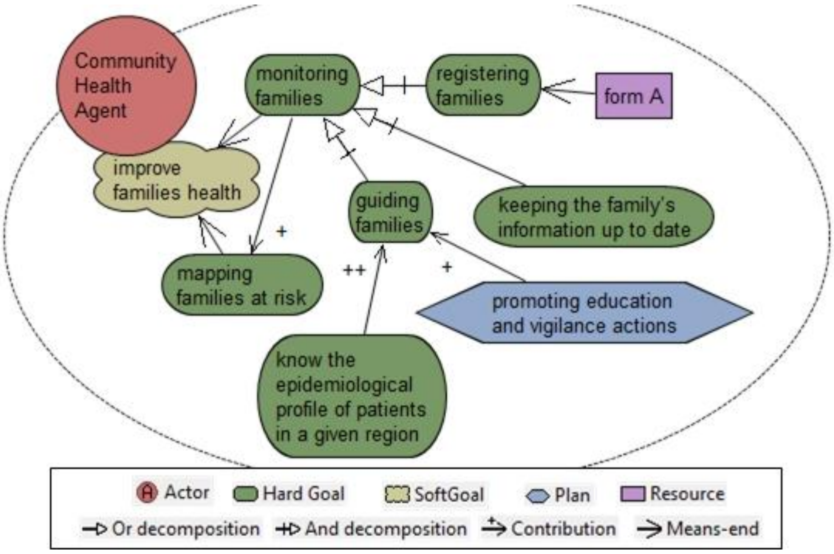


Fig. 2. Early requirements model focusing on the Community Health Agents

The diagram presented in Fig. 2 of the Tropos' early requirements activity Tropos, helps us understand how the Community Health Agent is included within the family health program (objective i of subsection 5.1), as well as their goals and their dependencies (objective ii of subsection 5.1).

6.3 Late Requirements: Modeling the Competency Questions

In this specific case study, we already started the work with the objective of developing an ontology-based system. So, the next step in our modeling process is to capture and model the Competency Questions to determine the scope of the ontology. The scope should define all the knowledge that should be in the ontology, as well as those that should not. The latter means that a concept should not be included, if there is not a competence question that uses it. This rule is also used to determine whether an axiom in the ontology should be included or not [Brusa et al. 2006].

Fig. 3 updates the model of Fig. 2 to include some competency questions that will later guide the tasks of the ontology engineer. As the reader may note, such competency questions' start being captured in the perspective of the organizational actor, in this case the Community Health Agent.

Table III textually presents the competency questions. Please note that in the model, for reasons of space, we just use codes, such as CQ1, CQ2 etc. to represent such questions. That is why to understand the model, the analyst generally needs this auxiliary table.

Table III. Competency Questions

CQ1.	What are the families at risk?
CQ2.	What are the families that receive more home visits?
CQ3.	What are the families who received no visit during a particular period in time?
CQ4.	What is the epidemiological profile of patients in a given region, i.e. age range, gender and ICD (International Classification of Diseases)?

For achieving the "*mapping families at risk*" goal, the Community Health Agent must have sufficient information to enable her to perform this mapping. For example, knowing what are the living conditions of families, the number of people living in the home, the health conditions of people in the family, are some of the information that the Community Health Agents needed to classify the risk of the family.

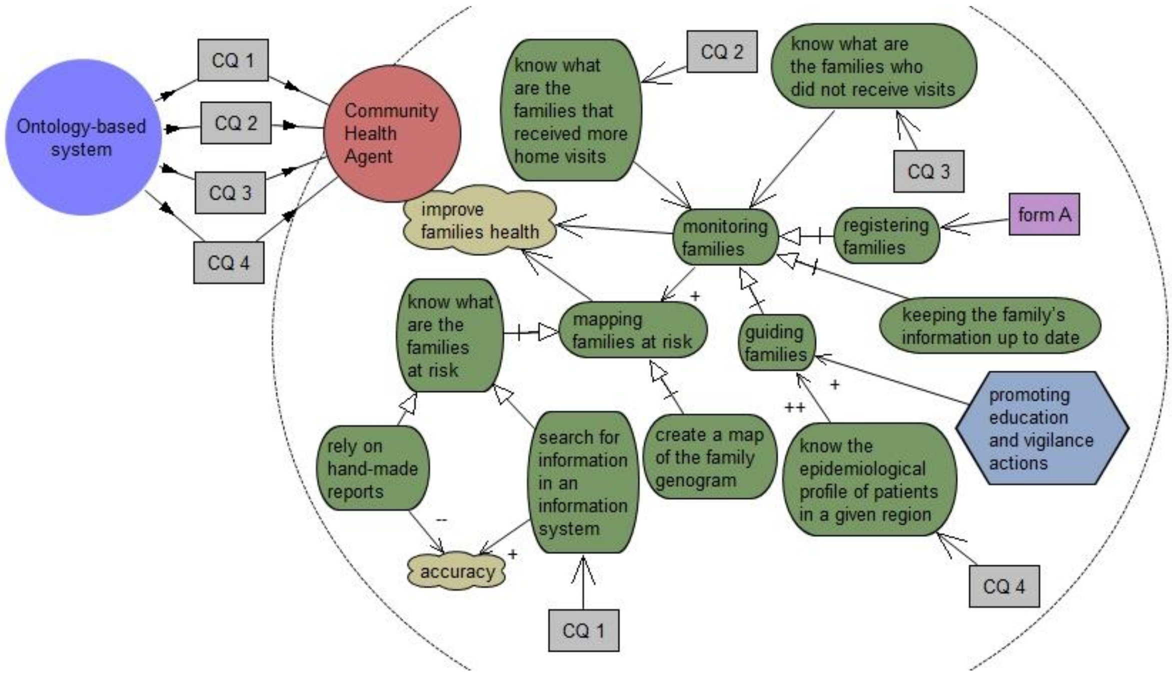


Fig. 3. Late Requirements model capturing and delegating the competency questions

To identify which families are at risk the Community Health Agent could "*rely on hand-made reports*" or "*search for information in an information system*" that will provide this support. When choosing to rely on hand-made reports, the Community Health Agent is increasing inaccuracy of the results, because reports on paper only become information management more difficult increasing the possibility of errors. Errors that could, for example, map incorrectly high-risk families, such as low-risk, leading to this family mapped incorrectly serious problems.

The fact of not properly perform the monitoring of a family, implies that only when something serious happens to a family member that the error is identified, in other words, would expect the problem to happen for later to be resolved, which is exactly contrary to the Family Health Program, where the main idea and promote the health of the population, not hoping the problem will occur for some action to take.

On the other hand, when the Community Health Agent uses an information system to get information on families at risk, he is increasing the accuracy of the results because the information system can quickly provide the information in different ways, facilitating the study by the Community Health Agent.

At this point **CQ 1** assists the Community Health Agent in its activity by providing necessary information for the purpose of "*mapping families at risk*" and also for this reason that the **CQ 1** is linked to the goal "*search for information in an information system*".

To support the achievement of "*monitoring families*", competency questions **CQ 2** and **CQ 3** assist the Community Health Agent, by providing information such as the families who have not yet been visited, and which of them are being more closely monitored. Since a Community Health Agent may be responsible for 500 people or more, this support is very important.

For accomplishing the "*know the epidemiological profile of patients in a given region*" goal, a lot of information must be gathered. This must be done in a systematic way and is the focus of **CQ 4**. And besides, a piece of information itself is not enough to provide the knowledge the Community Health Agent needs. For that, such information must be crossed with each other.

Still in the late requirements activity, we must model the system-to-be and its main goals. Fig. 3 shows the Ontology-based System actor, representing this system. And it also shows some resource dependencies between the system and the Community Health Agent. These resources are the competency questions that should be answered by the system (objective iii of subsection 5.1).

In Fig. 4, we model the perspective of the Ontology-based System actor. For simplicity, we only detail the goals of the system to support competency question **CQ 1**. It is important to note that although we are modeling the goals of the system, the competency questions refer to the ontology itself and not the system as a whole. Thus, the ontology is also represented as a resource within the system actor and a means-end link connects the competency questions to the ontology.

To accomplish the "*checking family*" goal, the system executes the "*classifying risk*", using the steps defined in a scale called the Scale of Rabbit [Coelho and Savassi 2004]. The goal is then decomposed in two sub-goals: "*checking health conditions*" and "*checking living conditions*".

Scale of Rabbit defines for each health or living condition which score should be given, the sum of scores defines the level of risk.

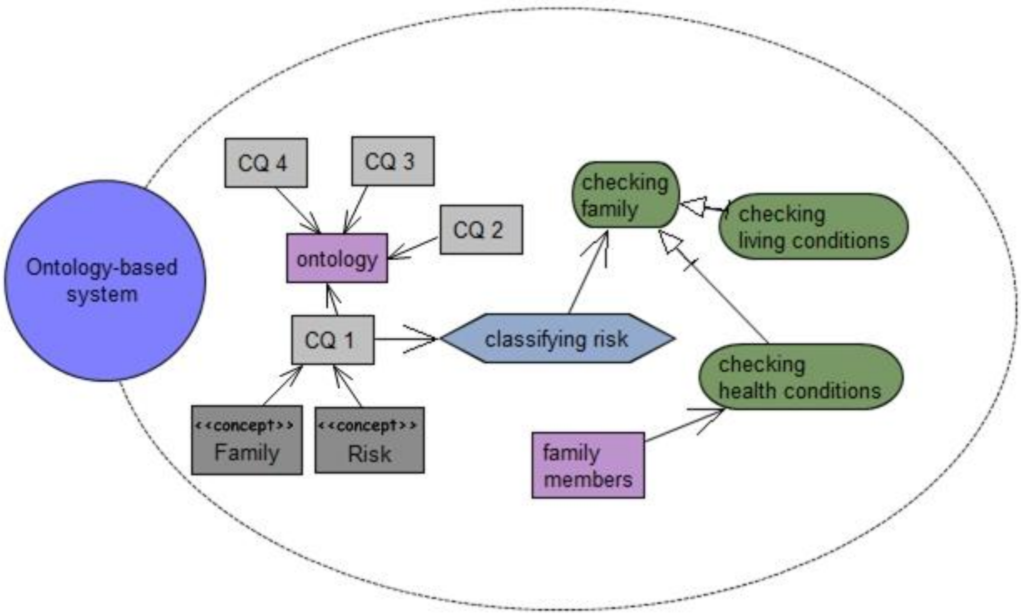


Fig. 4. Actor diagram of Ontology-based system

After capturing the competency questions, the analyst is already able to start eliciting a few concepts which will later compose the ontology that is the basis of the supporting system. For example, the "*Family*" and "*Risk*" concepts, linked to **CQ 1** illustrate how this may be modeled using Tropos. The idea here is not to be exhaustive, worrying about eliciting all concepts of the ontology. Capturing a few concepts that more obviously derive from the modeled competency questions provides a smooth transition to the following stage of ontology modeling.

6.4 Concepts for ontology

To develop the ontology, we must refine the concepts coming from the late requirements model. In this article, we only modeled two concepts related to **CQ 1** (see Fig. 4): "*Risk*" and "*Family*".

Fig. 5 shows a conceptual model of the ontology, showing how these two concepts may be refined. In this figure, we apply OntoUML [Guizzardi 2005], an Ontologically Well-Founded version of UML for conceptual modeling and domain ontology engineering. This language includes a number of distinctions proposed by the Foundational Ontology UFO, some of which appear as stereotypes in the model of Fig. 5.

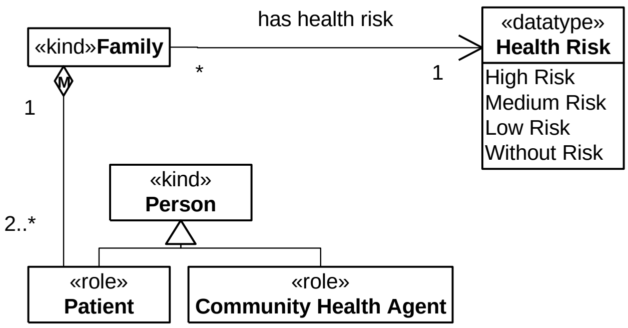


Fig. 5. The Ontology using OntoUML

Looking at Fig. 5, we observe a number of finer-grained distinctions among the represented UML classifiers. For instance, by stereotyping Family and Person as a Kind, the model indicates that these types provide a principle of identity for its instances and that the instances of those types are statically classified as instances of that type (e.g., an instance of Person cannot cease to be so without ceasing to exist). In contrast, by representing Patient and Community Health Agent as Role we have that every instance of those types can cease to be an instance of those types without ceasing to exist. Moreover, a Person plays the role of patient when being considered as part of a family whose health risk is being assessed; and plays the role of Community Agent when mapping family health risks. In both cases, the conditions for instantiating those types are relational conditions. As a consequence, the same person can be both a patient and a health community agent. Furthermore, a family has a risk associated with it and that risk (a property) can be measured in a given Risk Conceptual Space (represented in OntoUML by a Datatype). In this model, this conceptual space is composed of four ordered datatype values: without risk, high, medium and low.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the use of goal modeling (using Tropos) to capture and model the competency questions (i.e. the requirements) of an ontology. As far as we know, this is the first attempt to provide a visual solution to support these early stages of ontology engineering.

Here, we discuss the support provided by existing ontology-engineering methodologies and describe the proposed approach step-by-step. In addition to that, we also present a case study developed in a real health care setting, which illustrates and provides some interesting insights on the use of the methodology in practice.

Our idea is to provide a methodology which considers both the case in which the ontology is automated and the case in which it serves as a conceptual model to mediate and support people's communication and cooperation, without the need of a system. In this paper, we focused in the former, i.e. we model an ontology-based system.

Ontology reuse is one of the steps we intend to add in the methodology, allowing building an ontology in one subject reusing other different ontologies on that subject (*ontology merge*) or building an ontology in one subject reusing one or more ontologies in different subjects (*ontology integration*) [Pinto et al. 1999].

One of the most important topics in our future agenda concerns validation. Up to this point, we have developed the case study described in section 6, which worked as a proof of concept of our proposed methodology. Since this study has been developed in a real setting, it greatly helped us to understand the validity of our methodology, also providing us with interesting insights which led to improvements in our original ideas. However, to be thorough, we should develop a validation experiment in which we hope to compare our methodology with some existing approaches, by having real modelers apply them in a practical assignment.

Further research steps are needed to understand what kinds of adjustments should be made in the methodology to accomplish the latter.

Another research direction includes the development of the supporting system, using Semantic Web technologies. The city hall and the department of health of our city is interested in using this system to support their practical work related to the Family Health Program.

Besides the use of UFO to support the domain ontology development, our work also made another kind of use of such foundational ontology. As in previous initiatives, UFO was also applied to enable the methodology to be consistently extended, not hurting the semantics of the existing constructs. Moreover, we tried to add new elements having their own specific semantics (and thus, a new syntax as well). It is however necessary to reflect some more about a few of these constructs. For instance, it seems that we are making excessive use of the means-end link (it connects competency questions to goals, concepts to competency questions and concepts to the ontology resource). It remains to be seen if this relation should be specialized in different kinds of links to make the connection between these elements more clear.

ACKNOWLEDGMENT

This research has been partially supported by FAPES (Grant #45444080/09) and CNPq (Grant #481906/2009-6 and #483383/2010-4). The third author of this chapter holds the CNPq (Brazilian National Research Council) Productivity Grant #309382/2008-4. We are also grateful to the city hall and the department of health of Vitória/ES, Brazil, for all the support we had in conducting the case study.

REFERENCES

BERNARAS, A., LARESGOITI, I., CORERA, J. Building and reusing ontologies for electrical network applications, *in The European Conference on Artificial Inteligence, ECAI-96*, pp. 298-302, 1996.

BOEHM, B. *Industrial software metrics: top ten list*, IEEE, Software, Sept. 1987.

BRESCIANI, P., PERINI, A., GIORGINI, P., GIUNCHIGLIA, F. and MYLOPOULOS, J. Tropos: An Agent-Oriented Software Development Methodology Autonomous Agents and Multi-Agent Systems, *in International Journal of Autonomous Agents and Multi Agent Systems*, 2004, pp. 203–236.

BRUSA, G., CALIUSCO, M . and CHIOTTI, O. A process for building a domain ontology: an experience in developing a government budgetary ontology, *Proc. Second Australasian workshop on Advances in ontologies (AOW '06), v. 72*, Australian Computer Society, Inc., Darlinghurst, Australia, 2006. pp. 7-15.

CAMPOS, M. L. A. *O papel das definições na pesquisa em ontologia*, Perspect. ciênc. inf., Belo Horizonte, v. 15, n. 1, Apr. 2010.

COELHO, F. L. G. and SAVASSI, L. C. M. *Aplicação da Escala de Risco Familiar como instrumento de priorização das visitas domiciliares*, Revista Brasileira de Medicina de Família e Comunidade, Brasil, v. 1, n. 2, pp. 19-26, 2004.

DAVIS, A. M., *Software Requirements*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

FERNANDEZ, M., GOMEZ-PEREZ, A. and JURISTO, N. Methontology: from ontological art towards ontological engineering, *in Symposium on Ontological Engineering*, Proc. AAAI1997 Spring, 1997. pp. 33-40.

FILHO, N. A. and ROUQUAYROL, M. Z. *Introdução à Epidemiologia*, Rio de Janeiro: MEDSI, 2002, pp. 289.

FOX, M. The TOVE project towards a common-sense model of the enterprise, *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Springer Berlin / Heidelberg, pp. 25-34, 1992.

FRANCO, T. and MERHY, E. PSF: Contradições e novos desafios, *Conferência Nacional de Saúde On-Line*, Belo Horizonte / Campinas, Mar. 1999.

GASEVIC, D., DJURIC, D. and DEVEDZIC, V. *Model Driven Engineering and Ontology Development*, 2nd Edition, Springer, Berlin, 2009.

GÓMES-PÉREZ, A., FERNÁNDES-LÓPEZ, M.and CORCHO, O. *Ontological Engineering*, Springer-Verlag, New York, 2004, pp. 403.

GÓMEZ-PÉREZ, A. and SUÁREZ-FIGUEROA, M. C. Scenarios for building ontology networks within the NeOn methodology. *in Proceedings of the fifth international conference on Knowledge capture* (K-CAP '09). ACM, New York, NY, USA, pp. 183-184, 2009.

GRÜNINGER, M. and FOX, M.S. Methodology for the Design and Evaluation of Ontologies, *Proc. IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, August 19-20th, 1995.

GUARINO, N. Formal Ontology in Information Systems, *Proc. FOIS'98,* Amsterdam, IOS Press, pp. 3-15, 1998.

GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models*, PhD Thesis, University of Twente, The Netherlands. 2005.

GUIZZARDI, G. and WAGNER, G. Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages, *in Theory and Application of Ontologies*, ed.Berlin: Springer-Verlag, 2010.

GUIZZARDI, G., LOPES, M., BAIÃO, F. and FALBO, R. On the importance of Truly Ontological Distinctions for Ontology Representation Languages: An Industrial Case Study in the Domain of Oil and Gas, *In Proceedings of the 14th International Conference on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD´09)*, Amsterdam, The Netherlands, Lecture Notes in Business Information Processing (LNBIP), Springer-Verlag, 2009.

GUIZZARDI, R.S.S., LUDERMIR, P.G. and SONA, D. A Recommender Agent to Support Knowledge Sharing in Virtual Enterprises, *in Protogeros*, N. (Ed.). Agent and Web Service Technologies in Virtual Enterprises, Idea Group Publishing, 2007.

LEFEVRE, F. and LEFEVRE, A. M. C. *Promoção de Saúde: a negação da negação*, Vieira & Lent, São Paulo, 2007.

LENAT, D. B. CYC: a large-scale investment in knowledge infrastructure. *Commun. ACM* 38, pp. 33-38, Nov. 1995.

LENAT, D. B., GUHA, R. V., PITTMAN, K., PRATT, D. and SHEPHERD , M. Cyc: toward programs with common sense. *Commun. ACM* 33, pp. 30-49, Aug. 1990.

LIASKOS, S., MCILRAITH, S. A., SOHRABI, S. and MYLOPOULOS, J. Integrating Preferences into Goal Models for Requirements Engineering, *Proc. 10th International Requirements Engineering Conference (RE-10)*, 2010.

LUGER, G. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 5th edition, Boston, Addison-Wesley, 2005.

LUTZ, R. R. Analyzing Software Requirements Errors in Safety-Critical Embedded Systems, *Proc. ACM SIGSOFT Symposium on the Foundations of Software Engineering*, New York, NY, Dec. 1993, pp. 126-133.

NOY, F. N. and GUINNESS, D. L. *Ontology development 101: a guide to create your first ontology,* Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, 2001.

PINTO, H.S., GÓMEZ-PÉREZ, A., MARTINS, J. P. Some Issues on Ontology Integration, *Proc. of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, 1999.

SOARES, A., *Towards ontology-driven information systems: Guidelines to the creation of new methodologies to build ontologies*, Ph.D., The Pennsylvania State University, 137 pages, 2009.

SUÁREZ-FIGUEROA, M. C., DELLSCHAFT, K., MONTIEL-PONSODA, E., VILLAZÓN-TERRAZAS, B., YUFEI, Z., AGUADO DE CEA, G., GARCÍA, A., FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A., ESPINOZA, M. and SABOU, M. NeOn Deliverable D5.4.1. *NeOn Methodology for Building Contextualized Ontology Networks*. NeOn Project. http://www.neon-project.org. February 2008.

USCHOLD, M. and KING, M. Towards a methodology for building ontologies., *Proc. IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, July, 1995.

USCHOLD, M., KING, M., MORALEE, S., and ZORGIOS, Y. The Enterprise Ontology. *Knowl. Eng. Rev.* 13, pp. 31-89, Mar. 1998.