

A importância de Ontologias de Fundamentação para a Engenharia de Ontologias de Domínio: o caso do domínio de Processos de Software

G. Guizzardi, R.A. Falbo e R.S.S. Guizzardi

Resumo — Ontologias de Fundamentação (*Foundational Ontologies*) são sistemas de categorias filosoficamente bem fundamentados e independentes de domínio que têm sido utilizados com sucesso para melhorar a qualidade de linguagens de modelagem e modelos conceituais. Neste artigo, são apresentadas as últimas evoluções feitas em uma ontologia de fundamentação particular denominada UFO (*Unified Foundational Ontology*). Além disso, discute-se a relevância de ontologias de fundamentação no desenvolvimento de ontologias de domínio por meio de um estudo de caso no domínio de processos de software.

Palavras-chave — Ontologias em Engenharia de Software, Ontologias de Fundamentação, Processo de Software.

I. INTRODUÇÃO

Desde o final dos anos sessenta, tem-se reconhecido ontologias como um instrumento conceitual bastante útil na Ciência da Computação, principalmente nas áreas de modelagem de dados e inteligência artificial [1]. Nos últimos oito anos, ocorreu uma explosão de trabalhos relacionados a ontologias em diferentes áreas da Ciência da Computação, motivados, sobretudo, pelo crescente interesse na Web Semântica (*Semantic Web*) e no papel desempenhado por elas nessa iniciativa.

Um importante ponto a ser enfatizado é a diferença nos sentidos do termo ontologia quando usado em computação, de um lado, pela comunidade de Modelagem Conceitual e do outro pelas comunidades de Inteligência Artificial, Engenharia de Software e Web Semântica [1]. Em Modelagem Conceitual e áreas relacionadas (tal como Modelagem Organizacional), o termo tem sido usado de acordo com sua definição em Filosofia, a saber, como um sistema de categorias formais independente de domínio e filosoficamente bem fundamentado que pode ser usado para enunciar modelos da realidade específicos de domínio. Em contraste, nas demais áreas mencionadas, geralmente o termo ontologia é usado como: (i) um artefato concreto de engenharia projetado para um propósito específico sem dar muita atenção para questões de fundamentação, (ii) uma representação de um domínio particular (p.ex., biologia molecular, direito etc) expressa em

alguma linguagem de representação de conhecimento (p.ex., RDF, OWL, F-Logic) ou de modelagem conceitual (p.ex., UML, EER).

Ontologias, no sentido filosófico, têm sido desenvolvidas em Filosofia pelo menos desde Aristóteles e recentemente várias dessas teorias têm sido propostas na área de Ontologia Aplicada (*Applied Ontology*) sob o nome de Ontologias de Fundamentação (*Foundational Ontologies*). Neste artigo, discute-se uma ontologia de fundamentação particular denominada UFO (*Unified Foundational Ontology*).

UFO tem sido desenvolvida baseada em um número de teorias das áreas de Ontologias Formais, Lógica Filosófica, Filosofia da Linguagem, Linguística e Psicologia Cognitiva. O cerne dessa ontologia (UFO-A) é apresentado em detalhes e formalizado em [1]. Além disso, em outros trabalhos, UFO tem sido aplicada com sucesso para avaliar, (re)projetar e integrar os modelos de linguagens de modelagem conceitual, bem como para prover semântica de mundo real (*real-world semantics*) para seus elementos de modelagem (p.ex., [1-4]).

São dois os objetivos deste artigo. Primeiro, é apresentada uma nova versão de dois fragmentos de UFO, denominados UFO-B e UFO-C, inicialmente propostos em [4]. UFO-B é uma ontologia de eventos. UFO-C fundamenta-se em UFO-A e UFO-B para sistematizar conceitos sociais, tais como plano, ação, objetivo, agente, intencionalidade, comprometimento e compromisso. Segundo, trabalha-se a relação entre os dois sentidos de ontologias anteriormente mencionados, ilustrando a importância de ontologias de fundamentação no desenvolvimento de ontologias de domínio. Em particular, mostra-se como utilizar UFO para avaliar, re-projetar e dar semântica de mundo real para uma ontologia de domínio, usando como estudo de caso a ontologia de processo de software desenvolvida no Projeto ODE (*Ontology-based software Development Environment*) [5].

O restante deste artigo está organizado da seguinte forma: na seção II, apresenta-se um pequeno fragmento de UFO-A, discutindo apenas aquelas categorias que são essenciais para o entendimento das seções que se seguem; nas seções III e IV, são introduzidas as novas teorias desenvolvidas para UFO-B e C, respectivamente; a seção V apresenta uma parte da ontologia de processo de software original de ODE e mostra a avaliação e o re-projeto dessa ontologia quando interpretada em termos da UFO; finalmente, a seção VI apresenta as considerações finais deste artigo.

Esta pesquisa tem sido parcialmente apoiada pelas fundações de apoio a pesquisa FAPES (Projeto INFRA-MODELA e Bolsa DCR #37274554/2007) e FACITEC (Projeto MODELA).

II. UFO-A: UMA ONTOLOGIA DE OBJETOS

Uma distinção fundamental dessa ontologia é entre as categorias de **Indivíduo** (Particular) e **Universal** (Tipo). Indivíduos são entidades que existem na realidade, possuindo uma identidade única. Universais, por sua vez, são padrões de características que podem ser instanciados em um número de diferentes indivíduos.

Substanciais (*substantial*) são indivíduos existencialmente independentes. Exemplos incluem objetos mesoscópicos do senso comum, tais como uma pessoa, um cachorro, uma casa, Tom Jobim e Os Beatles. A palavra **Modo** (*Mode*), em contraste, denota a instanciação de uma propriedade. Um modo é um indivíduo que só pode existir em outros indivíduos e é dito ser **inerente a** esses indivíduos. Exemplos típicos de modos são uma cor, uma carga elétrica, um sintoma etc. Um importante traço que caracteriza todos os Modos é o fato deles só poderem existir em outros indivíduos. Por exemplo, uma carga elétrica só pode existir em algum condutor. Colocando de forma mais técnica, diz-se que um modo é *existencialmente dependente* de outros indivíduos [1]. A dependência existencial também pode ser usada para diferenciar modos intrínsecos e relacionais: **modos intrínsecos** (*intrinsic modes*) são dependentes de um único indivíduo (p.ex., uma cor, uma dor de cabeça, uma temperatura); **modos relacionais** (*relators*) dependem de vários indivíduos (p.ex., um emprego, um tratamento médico, um casamento).

Seguindo essa linha e levando em consideração a distinção de categorias de indivíduos e universais, há categorias de **universais substanciais** (*substantial universals*) e **universais de modo** (*mode universals*). Exemplos do primeiro incluem Maçã, Planeta e Pessoa. Exemplos do último incluem Cor, Carga Elétrica e Dor de Cabeça.

Uma tentativa de modelar a relação entre modos intrínsecos e sua representação em estruturas cognitivas humanas é apresentada na Teoria de Espaços Conceituais introduzida em [6]. Essa teoria é baseada na noção de **estrutura de qualidade** (*quality structure*). A idéia é que para diversos universais de propriedade que pode ser percebidos ou concebidos há uma estrutura de qualidade associada na cognição humana. Por exemplo, altura e massa são associadas com estruturas unidimensionais isomórficas à parte não-negativa da linha dos números reais. Outras propriedades como cor e sabor são representadas por estruturas multidimensionais. Em [6], a percepção ou concepção de uma propriedade intrínseca pode ser representada como um ponto na estrutura de qualidade. Esse ponto é denominado na literatura de **quale**. Estruturas de qualidade e qualia são, junto com conjuntos, números e proposições, exemplos de **indivíduos abstratos**.

Relações são entidades que aglutinam outras entidades. Na literatura de Filosofia, duas categorias amplas de relações são tipicamente consideradas, a saber **relações formais** e **materiais**. Relações formais acontecem entre duas ou mais entidades diretamente, sem nenhum outro indivíduo intervindo. Em princípio, a categoria de relações formais inclui aquelas relações que formam a superestrutura

matemática de nosso arcabouço, incluindo *dependência existencial*, *parte-de*, *subconjunto-de*, *instanciação*, dentre outras [1]. Relações materiais, por outro lado, possuem estrutura material por si próprias e incluem exemplos como *trabalhar em*, *estar matriculado em* ou *estar conectado a*. Enquanto, por exemplo, a relação formal entre Paulo e seu conhecimento *x* de Grego acontece diretamente e tão logo Paulo e *x* existam, para que aconteça uma relação material *ser tratado em* entre Paulo e uma unidade médica U-M, uma outra entidade precisa existir para mediar Paulo e U-M, neste caso um tratamento. Essas entidades são denominadas **modos relacionais** (*relators*) e são indivíduos com o poder de conectar (mediar) outros indivíduos. Assim, um tratamento médico conecta um paciente a uma unidade médica, uma matrícula conecta um estudante a uma instituição de ensino etc.

Suponha que João *está casado com* Maria. Neste caso, assume-se que há um modo relacional m_1 do tipo *casamento* que media João e Maria. Além disso, há várias propriedades que João adquire em virtude de estar casado com Maria, bem como responsabilidades legais que João assume no contexto dessa relação. Essas novas propriedades adquiridas são instanciadas em modos intrínsecos de João que, por conseguinte, são existencialmente dependentes dele. Entretanto, esses modos também dependem da existência de Maria. Esse tipo de modo é chamado de **modo externamente dependente**, i.e, modos intrínsecos que são inerentes a um único indivíduo, mas que são existencialmente dependentes de outros (possivelmente vários) indivíduos. O modo relacional casamento, neste caso, é a soma de todos os modos externamente dependentes que João e Maria adquirem em virtude de estarem casados um com o outro.

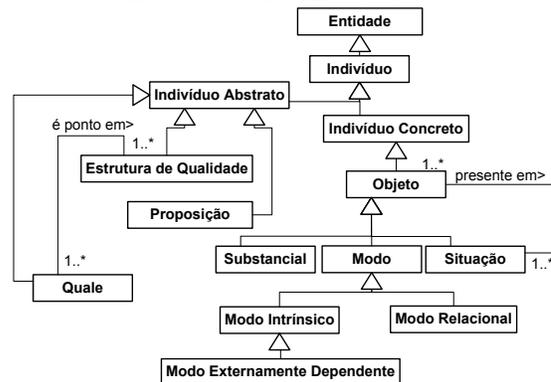


Fig. 1. Fragmento da UFO-A: Indivíduos

Finalmente, considera-se aqui a noção de **situação**. Situações são entidades complexas constituídas possivelmente por vários objetos (incluindo outras situações), sendo tratadas aqui como um sinônimo para o que é chamado na literatura de estado de coisas (*state of affairs*), ou seja, uma porção da realidade que pode ser compreendida como um todo. São exemplos de situações: “*João estar gripado e com febre*”, e “*Maria estar casada com João que trabalha para a UFES*”. Tomando por base a noção de situação, define-se a relação “*estar presente em*” entre objetos e as situações que eles constituem. Por exemplo, pode-se dizer que o substancial João

e seus modos febre e gripe estão presentes na situação “*João estar gripado e com febre*”. Para uma discussão mais detalhada de nossa visão de situações, veja [7].

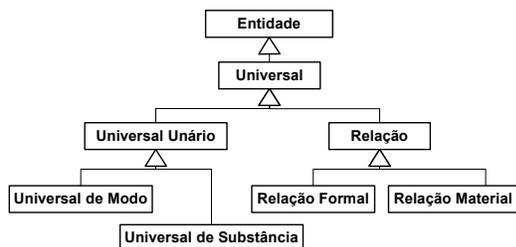


Fig. 2. Fragmento da UFO-A: Universais

III. UFO-B: UMA ONTOLOGIA DE EVENTOS

UFO-B diferencia explicitamente Eventos e Objetos (Figuras 3-5). Classicamente, esta distinção é feita em termos de suas respectivas relações com o tempo. Diz-se que objetos estão inteiramente presentes em qualquer instante do tempo que estiverem presentes, i.e., eles *são no tempo* no sentido de, se em uma circunstância c_1 , um objeto O possui a propriedade P_1 e em uma circunstância c_2 a propriedade P_2 (possivelmente incompatível com P_1), ele continua sendo o mesmo objeto O em ambas as situações. Por exemplo, pode-se dizer que o indivíduo João pesa 80Kg em c_1 e 70Kg em c_2 , mas ele é o mesmo indivíduo João.

Eventos (ou ocorrências) são indivíduos compostos de partes temporais. Eles *acontecem no tempo* no sentido de se estenderem no tempo acumulando partes temporais. São exemplos de eventos: uma conversa, uma partida de futebol, a execução de uma sinfonia e um processo de negócio. Em qualquer momento em que um evento está presente, apenas algumas de suas partes temporais estarão presentes. Como uma consequência, eventos não podem sofrer mudanças no tempo no sentido genuíno, uma vez que nenhuma de suas partes temporais mantém sua identidade ao longo do tempo.

Eventos são possíveis transformações de uma situação para outra na realidade, i.e., eles podem alterar o estado de coisas da realidade de um (pré)estado para outro (pós-estado). Eventos são entidades ontologicamente dependentes no sentido de, para existirem, dependerem existencialmente de seus participantes. Seja o evento e : *o ataque de Brutus a César*. Nesse evento, há a participação de César, Brutus e da faca usada no ataque. Neste caso, e é composto da participação individual de cada uma dessas entidades. Cada uma dessas participações é por si própria um evento que pode ser complexo ou atômico, mas que existencialmente depende de um único substancial. Em UFO-B, ser atômico e ser instantâneo são noções ortogonais, i.e., participações atômicas podem se estender no tempo, bem como eventos instantâneos podem ser compostos de múltiplas participações (instantâneas). Em suma, o modelo da Figura 3 mostra essas duas perspectivas sob as quais eventos podem ser analisados, a saber: como entidades que se estendem no tempo com certas estruturas mereológicas (i.e., eventos simples ou complexos), e como entidades ontologicamente dependentes que podem envolver um número de participações individuais.

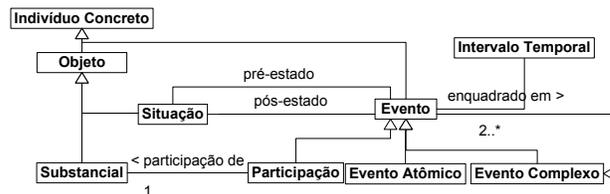


Fig. 3. Fragmento da UFO-B: Objetos e Eventos.

Considera-se nesta ontologia que todos os modos espaciais (p.ex. localização no espaço) de eventos são definidas em termos dos modos espaciais de seus participantes. Em contraste, todos os modos temporais de substanciais são definidos em termos dos eventos nos quais eles participam. Análogo ao que foi discutido para objetos, os modos temporais de eventos têm seus valores (*qualia*) obtidos pela sua projeção em uma estrutura de qualidade. Assume-se aqui, ainda, que o espaço conceitual de tempo é uma estrutura composta de **intervalos temporais** (*time intervals*) que, por sua vez, são compostos de **instantes** (*time point*). Instantes podem ser representados como números reais e intervalos temporais como conjuntos de números reais. Entretanto, eles também podem ser interpretados como entidades *sui generis*, tais como cronóides (*chronoids*) e fronteiras temporais [8]. Em outras palavras, nesse ponto, evita-se fazer comprometimentos ontológicos desnecessários. Em particular, este modelo permite uma diversidade de estruturas temporais, tais como tempo linear, ramificado, paralelo e circular. Para o caso de estruturas ordenadas, consideram-se as ditas *Relações entre intervalos de Allen* [9] a partir das quais as correspondentes relações entre eventos podem ser derivadas (Figures 4 e 5).

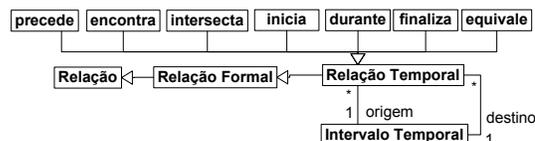


Fig. 4. Fragmento da UFO-B: Relações de Allen [9].

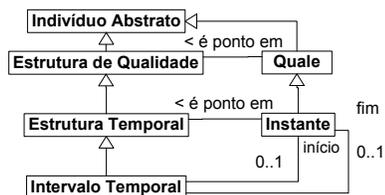


Fig. 5. Fragmento da UFO-B: Intervalos Temporais e Instantes.

IV. UFO-C: UMA ONTOLOGIA DE ENTIDADES SOCIAIS

A terceira camada da UFO é uma ontologia de entidades sociais (tanto objetos quanto eventos), construída sobre UFO-A e UFO-B. Fragmentos dessa ontologia são mostrados nas Figuras 6 e 7. Começa-se pela distinção entre substanciais agentivos e não agentivos, respectivamente denominados aqui por **agentes** e **substanciais inanimados**. Agentes podem ser físicos (p.ex., uma pessoa) ou sociais (p.ex., uma organização, uma sociedade). Substanciais Inanimados também podem ser indivíduos físicos (p.ex., um livro, um carro, uma árvore) ou sociais (p.ex., dinheiro, linguagem e normas). Uma **descrição**

normativa (*normative description*) [10] é um tipo de substancial inanimado social que define uma ou mais regras/normas reconhecidas por, pelo menos, um agente social e que pode definir entidades sociais como universais (p.ex., tipos de compromettimentos sociais), outros objetos (a coroa do rei da Espanha) e papéis sociais, tais como presidente, ou pedestre. São exemplos de descrições normativas a Constituição Brasileira e o regimento do Mestrado em Informática da UFES, bem como um conjunto de diretivas sobre como realizar algumas ações dentro de uma organização.

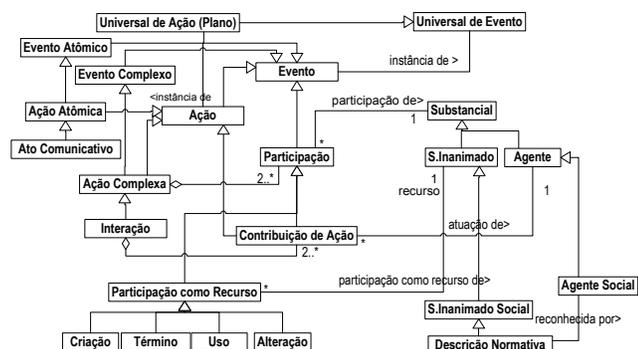


Fig. 6. Fragmento de UFO-C: Ações, Agentes e Substanciais Inanimados

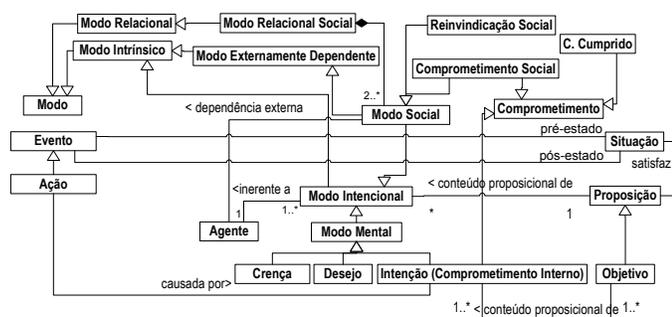


Fig. 7. Fragmento de UFO-C: Modos Mentais e Sociais

Agentes são substanciais que podem possuir tipos especiais de modos chamados de **modos intencionais** (*intentional modes*). Conforme sustentado em [11], intencionalidade deve ser entendida em um contexto mais amplo do que a noção de “alguma coisa que se intenciona”, mas como a capacidade de certas propriedades de certos indivíduos de se referir a possíveis situações na realidade. Todo modo intencional tem um tipo – p.ex, **crença** (*belief*), **desejo** (*desire*), **intenção** (*intention*) – e um conteúdo proposicional, sendo este último uma representação abstrata de uma classe de situações referenciadas por esse modo intencional. Assim, “alguma coisa que se intenciona” é um tipo específico de intencionalidade denominada **intenção** (*intention*). O conteúdo proposicional de uma intenção é um **objetivo** (*goal*). A relação precisa entre um modo intencional e uma situação é a seguinte: uma situação na realidade pode satisfazer o conteúdo proposicional de um modo intencional (i.e., satisfazer, no sentido lógico, a proposição representando o conteúdo proposicional). Crenças podem ser justificadas por situações na realidade. Exemplos incluem a crença de que Roma é a capital da Itália e de que a Lua move-se em órbita

em torno da Terra. Desejos e intenções podem ser realizados ou podem ser frustrados. Enquanto um desejo expressa uma vontade de um agente em direção a um estado de coisas na realidade (p.ex., o meu desejo de que o Brasil ganhe a próxima Copa do Mundo), intenções são estados de coisas desejados e que o agente se compromete a perseguir (comprometimento interno), tal como a intenção de passar as próximas férias na praia [11] [12]. Por essa razão, intenções fazem com que agentes executem **ações**.

Ações são eventos intencionais, i.e., eventos que instanciam um **plano** (ação universal) com o propósito específico de satisfazer (o conteúdo proposicional de) alguma intenção. São exemplos de ações: escrever este artigo, um processo de negócio, um ato comunicativo [11] etc. Assim como eventos, ações podem ser atômicas ou complexas. Já uma ação complexa é composta de duas ou mais participações. Essas participações, por sua vez, podem ser intencionais (sendo, portanto, elas próprias ações) ou eventos não intencionais. Por exemplo, o ataque de Brutus a César inclui a participação intencional de Brutus e a participação não intencional da faca. Em outras palavras, seguindo a tradição de teorias filosóficas de ação (*action theories*), nem toda participação de um agente é considerada uma ação, mas somente as participações intencionais, aqui denominadas **contribuições de ação** (*action contributions*). Apenas agentes (entidades capazes de possuir modos intencionais) podem realizar ações.

Um substancial inanimado participando em uma ação é denominado um **recurso**. Uma ação complexa composta de contribuições de ações de diferentes agentes é denominada uma **interação** (*interaction*). Dois artistas colaborando para criar uma escultura é um exemplo de uma interação, assim como um diálogo entre dois agentes. No primeiro caso, a escultura, as ferramentas e o material usados para criá-la são exemplos de recursos. Substanciais inanimados podem participar em ações de diferentes maneiras. Neste artigo, são considerados quatro tipos de **participação de recurso** (*resource participation*), a saber: **criação** (*creation*), **término** (*termination*), **alteração** (*change*) e **uso** (*usage*), que podem ser caracterizados como se segue. Sejam r um recurso, a uma ação e s_1 e s_2 duas situações correspondendo, respectivamente, ao pré e ao pós-estado da ação a . Tem-se, então:

- criação: uma participação de recurso de r em a é uma criação sse: (r não está presente em s_1) E (r está presente em s_2) E (existe pelo menos uma contribuição de ação ac que é parte de a e que s_2 satisfaz o conteúdo proposicional de ac).
- término: uma participação de recurso de r em a é um término sse: (r está presente em s_1) E (r não está presente em s_2) E (existe pelo menos uma contribuição de ação ac que é parte de a e que s_2 satisfaz o conteúdo proposicional de ac).
- alteração: uma participação de recurso de r em a é uma alteração sse: (i) existe pelo menos um modo m tal que (m é inerente a r em s_1 E m não é inerente a r em s_2) OU (m não é inerente a r em s_1 E m é inerente a r em

s_2); E (ii) existe pelo menos uma contribuição de ação ac que é parte de a e que s_2 satisfaz o conteúdo proposicional de ac .

- uso: uma participação de recurso que não é de nenhum dos tipos acima é uma participação de uso.

Conforme discutido em [3], uma participação de recurso pode ser a causa de uma *dependência de recurso* e o resultado de uma *aquisição de recurso* entre agentes. Em uma aquisição de recurso de um agente B para um agente A , A dá permissão do recurso r para B . Para isso acontecer, A deve ter o direito de conceder permissão para o agente B e, além disso, o direito de conceder o direito do tipo de permissão (p.ex, usar, alterar). Em outras palavras, diferentes tipos de participações de recurso podem ser ligados a diferentes conseqüências deonticas da relação entre agentes. Em suma, uma participação de recurso em uma ação corresponde à criação (alteração, término ou uso) pretendida de um objeto nessa ação. Assim, considera-se, por exemplo, uma alteração do recurso r na ação a quando essa alteração é o conteúdo da intenção de pelo menos um dos agentes nessa ação.

Atos comunicativos podem ser usados para criar **modos sociais** (*social modes*). Nessa visão, a linguagem não somente representa a realidade mas também cria parte dela [11]. Assim, propriedades sociais são tipos de propriedades intencionais que são criados pela troca de atos comunicativos e pelas conseqüências dessas trocas (p.ex., adoção de uma objetivo, delegação etc [3][12]). Suponha que João alugue um carro em uma locadora de automóveis. Quando ele assina um contrato de locação, ele está executando um ato comunicativo, no caso uma promessa. Esse ato cria um **comprometimento social** (*social commitment*) com a locadora: o comprometimento de *devolver o carro em certas condições* etc (o conteúdo proposicional). Além disso, cria também uma **reivindicação social** (*social claim*) da locadora para com João, relativa a esse conteúdo proposicional. Um **modo relacional social** (*social relator*), ou **vínculo social** (*social bond*), é um tipo de modo social composto de dois ou mais pares de comprometimentos/reivindicações (propriedades sociais) associados. Finalmente, um comprometimento (interno ou social) é cumprido por um agente se esse agente realiza uma ação tal que o pós-estado dessa ação é uma situação que satisfaz o conteúdo proposicional daquele comprometimento.

A. Distinção entre Universais de Ação, Ocorrências de Ação e Ações Programadas

Nesta subseção, elaboram-se alguns dos conceitos discutidos anteriormente com a intenção de clarificar uma distinção que é comumente obscura em diversos modelos de processo, a saber a distinção entre universais de ação (*action universals*), ocorrências de ações (*action occurrences*) e ações programadas (*scheduled actions*). A Figura 8 apresenta categorias ontológicas adicionais da UFO-C necessárias para discutir essas distinções.

Inicia-se com um detalhamento da noção de comprometimento (*commitment*). Comprometimentos (internos ou sociais) podem ser **cumpridos** (*fulfilled*) ou **não**

cumpridos (*unfulfilled*). Comprometimentos não cumpridos podem estar **pendentes** (*pending*), **desmarcado** (*dismissed*) ou **quebrados** (*broken*). Um **comprometimento social** (*social commitment*) C é um comprometimento de um agente A para com outro agente B . Na qualidade de um modo externamente dependente, diz-se que C é inerente a A e externamente dependente de B . Diferentemente de um comprometimento interno, neste caso, C só pode ser desmarcado por B . **Comprometimentos internos** ou **intenções** (*intentions*) fazem com que o agente realize ações. Assim, seguindo [12], tem-se que comprometimentos sociais necessariamente causam a criação de comprometimentos internos, i.e., se Pedro promete trazer um livro amanhã para João, além do comprometimento com João, ele também tem a intenção (comprometimento interno) de trazer o livro amanhã.

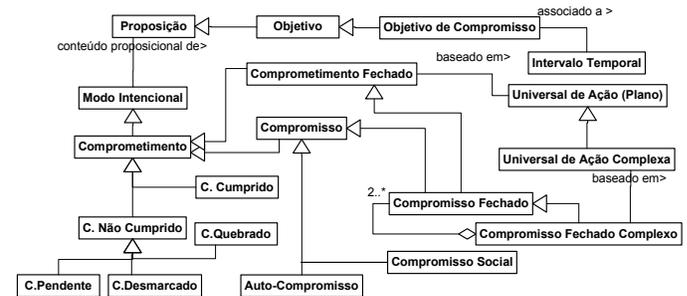


Fig. 8. Fragmento da UFO-C tratando de Compromissos.

Como discutido na seção anterior, um comprometimento é cumprido por um agente A se esse agente realiza uma ação tal que o pós-estado dessa ação é uma situação que satisfaz o conteúdo proposicional do comprometimento. Em muitos casos, há diversas ações que podem levar a essa situação. Diferenciam-se **comprometimentos abertos** (*open commitments*) e **fechados** (*closed commitments*), sendo a diferença que, no caso do último, o agente deve cumprir o comprometimento pela execução de uma ação específica. Diz-se, então, que um comprometimento C é baseado em um plano P tal que C é cumprido pelo agente A se e somente se A ativamente provocar uma situação S que satisfaz o conteúdo proposicional de C e por meio de uma ação p' que é uma instância de P . Comprometimentos abertos e fechados podem explicar as noções de delegação aberta e fechada [12], respectivamente. Essas, por sua vez, podem explicar a diferença entre o que é chamado de Dependência de Objetivo (*Goal Dependence*) e Dependência de Plano (*Plan Dependence*) em linguagens de Engenharia de Requisitos, tais como TROPOS e i* [3].

Um tipo especial de comprometimento é um **compromisso** (*appointment*). Um compromisso é um comprometimento cujo conteúdo proposicional explicitamente refere-se a um intervalo temporal. Por exemplo, enquanto “Pedro promete devolver o livro para João” é um comprometimento social, “Pedro promete devolver o livro para João nessa semana” é um compromisso. Um compromisso pode ser interno, denominado aqui de **auto-compromisso** (*self-appointment*), ou um **compromisso social** (*social appointment*). Um **compromisso fechado** (*closed appointment*) é um

comprometimento fechado cujo conteúdo proposicional se refere explicitamente a um intervalo temporal. Um **compromisso fechado complexo** (*complex closed appointment*) é baseado em uma **universal de ação complexa** (*complex action universal*). De um lado, um compromisso fechado complexo *C* é composto de vários comprometimentos que devem ser satisfeitos pela execução de um número de ações que são parte de uma ação complexa (instância de um plano complexo no qual *C* é baseado). De outro, *C* representa um conjunto de comprometimentos de executar uma instância de um plano complexo pela execução de suas sub-ações específicas (i.e., criando instâncias dessas ações) dentro de intervalos temporais específicos (referenciados pelo conteúdo proposicional do seus sub-comprometimentos).

Neste ponto, é possível tornar claro que ações programadas (*scheduled actions*) não são nem ocorrências de ações (*action occurrences*) (i.e., eventos particulares que ocorrem em intervalos de tempo específicos) nem universais de ação (*action universals*) (padrões de características instanciados por múltiplas ocorrências de ações). De fato, uma ação programada não é uma ação. Ao contrário, é um comprometimento de instanciar um plano em um intervalo de tempo específico, ou seja, é um compromisso fechado (*closed appointment*).

V. ANALISANDO E APERFEIÇOANDO UMA ONTOLOGIA DE PROCESSO DE SOFTWARE EM TERMOS DA UFO

Em [13], Falbo e Bertollo apresentam uma Ontologia de Processo de Software que foi desenvolvida para estabelecer uma conceituação comum para organizações de software falarem sobre processos de software. Essa ontologia é usada como base para o desenvolvimento de uma infra-estrutura de processo para ODE (*Ontology-based software Development Environment*) [5], um Ambiente de Desenvolvimento de Software Centrado em Processo. A Figura 9 mostra um fragmento dessa ontologia.

Conforme mostrado nessa figura, um *processo de software* pode ser decomposto em *atividades* ou outros processos, chamados de *sub-processos*. Uma atividade é uma porção de trabalho que pode produzir *artefatos* (*produtos*). Para ser realizada, uma atividade requer *recursos*, adota *procedimentos* e utiliza artefatos produzidos por outras atividades (*insumos*). Uma atividade pode ser decomposta em *sub-atividades* e pode depender da realização de outras atividades, ditas *pré-atividades*. Artefatos podem ser decompostos em *sub-artefatos*.

Recursos são usados durante, ou para apoiar, a execução de atividades. Esses recursos podem ser agrupados em três categorias principais: (i) *recursos humanos* são os papéis que agentes humanos devem desempenhar em uma atividade, tais como engenheiro de requisitos, gerente de projeto, cliente etc; (ii) *recursos de hardware* incluem quaisquer equipamentos de hardware requeridos para a realização de uma atividade, tais como computadores e impressoras; (iii) *recursos de software* dizem respeito a qualquer produto de software que é usado na realização de uma atividade, tal como uma ferramenta CASE

ou um sistema de gerenciamento de banco de dados.

Procedimentos são adotados na realização de atividades. Há diversos tipos de procedimentos. *Modelos de Documento*, por exemplo, são modelos a serem seguidos em uma atividade para a elaboração de um artefato. Um *método* é um procedimento sistemático que define um fluxo (*workflow*) de atividades (uma seqüência de passos) e heurísticas para realizar uma ou mais atividades. Quando um método pode ser adotado na realização de mais de uma atividade, ele possui um fluxo de trabalho (*workflow*) para cada uma delas.

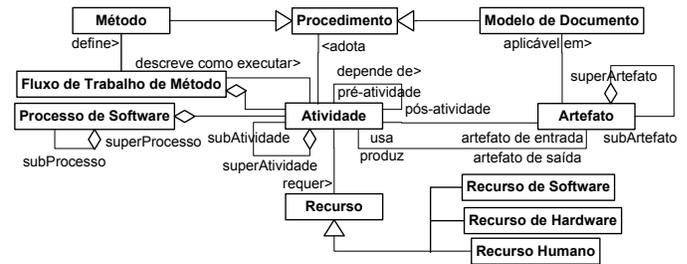


Fig. 9. Fragmento da Ontologia de Processo de Software de ODE.

As Figuras 10 e 11 representam fragmentos da ontologia de processo de software revisado, obtidos pela interpretação dos conceitos da ontologia original em termos de UFO-C. Nesses modelos, os conceitos da UFO são mostrados em cinza. Nessa interpretação, torna-se claro que a ontologia de processos do ODE funde as noções de universal de ação (*action universal*) e ocorrência de ação (*action occurrence*). Fez-se, portanto, a separação desses conceitos, introduzindo-se os termos Ocorrência de Atividade e Ocorrência de Processo de Software para denotar ações particulares que ocorrem em intervalos de tempo específicos. Uma ocorrência de atividade pode ser atômica ou complexa. Uma ocorrência de processo de software, em contraste, é necessariamente uma ação complexa. De fato, a distinção entre a ocorrência de um sub-processo (i.e., um processo que é parte de outro processo) e a ocorrência de uma atividade complexa não é clara na ontologia de domínio original. Neste trabalho, para eliminar essa ambigüidade, assume-se que uma ocorrência de processo de software é a raiz de um reticulado de composição, i.e, uma ocorrência de processo de software é uma ação complexa que não é parte de nenhum outro evento complexo. Uma consequência dessa definição é que não é possível haver sub-processos e, portanto, a relação todo-parte reflexiva entre ocorrências de processo de software foi removida do modelo. Em suma, uma Ocorrência de Processo de software é uma instância de Processo de Software que, por sua vez, é um subtipo de Universal de Ação Complexa. Uma Ocorrência de Atividade é uma instância de uma Atividade, que é um subtipo de Universal de Ação (Plano).

Na ontologia original, um artefato é um tipo de Substancial Inanimado. A relação de sub-artefato entre artefatos é, portanto, governada pelos axiomas mereológicos definidos para relações todo-parte entre substanciais definidos em [1]. Embora isso não possa ser detalhado aqui, tem-se que, geralmente, essas relações entre objetos são irreflexivas, assimétricas e não transitivas. Relações todo-parte entre

processos, em contraste, são relações de ordem parcial estrita.

A noção de recurso na ontologia do ODE pode ser mapeada para a noção de substancial na UFO-A e a relação *requer* sub-juíza diferentes tipos de participação de uma ocorrência de atividade. Acredita-se que essa questão precisa ser elaborada nessa ontologia, de modo a fazer justiça à distinção entre contribuições de ação e participação de recursos em UFO-C. Em particular, um recurso humano (um Agente em UFO-C) não pode ser usado, modificado, criado ou eliminado por uma ocorrência de atividade. Ao contrário, uma contribuição de ação de um recurso humano, na verdade, denota um comprometimento social desse agente em realizar parte dessa ocorrência de atividade (com suas permissões e obrigações decorrentes). Em termos de relações formais, em [3] apontamos que a relação *requer* para o caso de recursos humanos é um tipo de relação de dependência entre agentes, que conduz a uma relação de delegação quando o processo é instanciado ou programado. Em poucas palavras, um agente *A* depende de um agente *B* se e somente se *A* possui um objetivo *O* que ele não pode atingir sozinho (seja por falta de capacidade, seja pelo fato de *O* contrastar com um de seus outros objetivos) e *B* pode atingir *O*. O agente *A* delega o objetivo *O* para *B* se e somente se *A* depende de *B* em relação a *O* e *B* compromete-se em atingir *O* para *A*.

Recursos de hardware e de software são tipos de substanciais inanimados e, portanto, seus tipos de participação devem corresponder aos tipos de participação de recurso definidos na UFO-C. Nesse caso específico, a participação deve ser de uso. Por outro lado, um artefato de saída (produto) é um recurso com participação de criação em uma ocorrência de atividade, enquanto um artefato de entrada (insumo) é um recurso com participação de uso.

Vale ressaltar que um recurso em UFO-C é um papel que um objeto desempenha em um evento. Por conseguinte, o mesmo deve acontecer para suas subclasses: artefato de entrada, artefato de saída e recursos devem ser papéis desempenhados por substanciais inanimados no escopo de uma ocorrência de atividade. Como uma consequência, um substancial *S* que é um artefato de entrada (insumo) para uma ocorrência de atividade pode ser um artefato de saída de outra ocorrência de atividade. Agora, pode um substancial *S* que é um artefato de saída (produto) para uma ocorrência de atividade ser um recurso de software usado por outra ocorrência de atividade? Em outras palavras, uma vez que recursos humanos não são mais considerados recursos, o que caracteriza os diferentes papéis de produto e recurso, uma vez que ambos são desempenhados por substanciais inanimados com participação de uso? A resposta é a seguinte: se um objeto desempenha o papel de recurso em uma ocorrência de atividade, então não há uma participação de criação desse objeto em outra ocorrência de atividade que é parte da mesma ocorrência de processo de software.

Um procedimento é um tipo de Descrição Normativa. Em UFO-C, há dois tipos especiais de descrições normativas: descrições de objeto e de plano. Uma **descrição de substancial inanimado** é uma descrição de um **universal de**

substancial inanimado, tal como a planta de uma casa. De maneira análoga, uma **descrição de plano** (*plan description*) é uma descrição de um **plano** (universal de ação), tal como diretivas sobre como montar uma cadeira. Um método é um tipo de descrição de plano, enquanto um modelo de documento é um tipo de descrição de substancial.

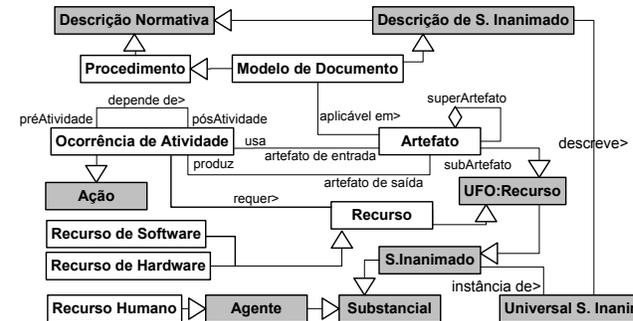


Fig. 10. Fragmento Remodelado da Ontologia da Processos de Software (Substanciais)

Uma outra questão na ontologia original é a relação de dependência entre atividades (agora, ocorrências de atividades). Dependência entre ocorrências de atividades é definida em termos de recursos e, mais especificamente, de artefatos: uma ocorrência de atividade *a* depende de uma ocorrência de atividade *b* se e somente se *a* utiliza como insumo um recurso produzido por *b*. Entretanto, os nomes de papéis usados nessa relação são pré-atividade e pós-atividade, indicando que há uma relação de ordenação escondida entre essas ocorrências de atividade. A que relações de ordenação temporal essas relações corresponderiam? Pela ontologia original, não é possível chegar a uma resposta. Em geral, as duas únicas relações temporais que podem ser descartadas de início são **termina** (*finishes*) e **equivalente** (*equals*), i.e., se uma ocorrência de atividade *a* depende de alguma coisa produzida por uma ocorrência de atividade *b*, então *a* não pode terminar simultaneamente ou antes de *b*. Assim, considerando relações de ordenação entre ocorrências de atividade instantâneas, a dependência implica que *b* ocorre antes de *a* (vide UFO-B), uma vez que as relações **inicia** (*starts*) e **durante** (*during*) não são definidas para instantes e a relação **encontra** (*meets*) degenera-se para **equivalente** neste caso. Entretanto, para o caso de ocorrências de atividade que se estendem no tempo, há quatro possibilidades: **precede** (*before*), **encontra** (*meets*), **inicia** (*starts*) e **durante** (*during*). Em qualquer caso, essas quatro relações são de ordem total e estrita (irreflexiva, assimétrica e transitiva) se um modelo linear de tempo for assumido. Em suma, uma vez que a dependência entre ocorrências de atividade implica uma relação de ordem total mais uma dependência de recurso, como resultado tem-se que a relação deve obedecer uma relação de ordem parcial (i.e., irreflexiva, assimétrica e transitiva, mas não totalmente definida).

Podem se tornar mais claras as duas relações entre método e fluxo de trabalho na Figura 9, fazendo-se uma distinção explícita entre Universal de Ação (*Action Universal*) e uma descrição desse universal. Um fluxo de trabalho de um método é um exemplo de uma descrição. Assim, não se pode

dizer que ele é composto de atividades. Como uma **descrição de plano** (*plan description*), um fluxo de trabalho de um método descreve como realizar um universal de ação complexa (um processo de software ou uma atividade complexa). A relação “descreve como realizar” é, portanto, uma especialização da relação “descreve” entre **Plano** (*Plan*) e **Descrição de Plano** (*Plan Description*). Além disso, como uma descrição de um universal de ação complexa, um fluxo de trabalho de um método descreve como realizar duas ou mais ações que compõem as instâncias desse universal.

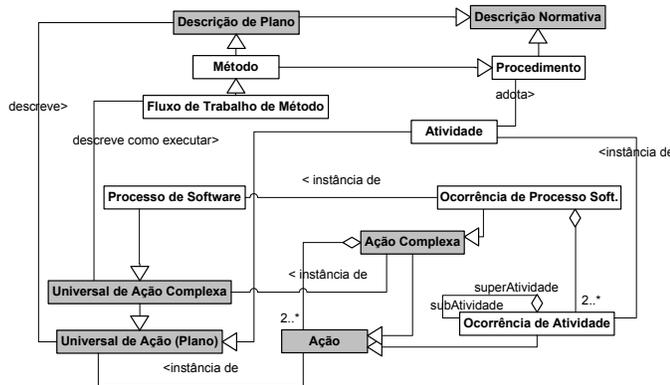


Fig. 11. Fragmento Remodelado da Ontologia da Processos de Software (Eventos)

VI. CONSIDERAÇÕES FINAIS

Este artigo apresentou as evoluções mais recentes da ontologia de fundamentação UFO. Em particular, foram discutidas as novas versões de dois fragmentos da UFO, a saber UFO-B (que diz respeito a eventos) e UFO-C (que trata de conceitos sociais e intencionais). Além disso, ilustrou-se o papel desempenhado por uma ontologia filosoficamente bem fundamentada no projeto de ontologias de domínio. Em particular, mostrou-se como UFO pode ser usada para avaliar, re-projetar e dar semântica de mundo real para uma ontologia no domínio de engenharia de software, a saber a ontologia de processo de software que é o cerne do ambiente de desenvolvimento de software ODE. Fazendo isso, corrigiu-se um número de problemas conceituais nessa ontologia, tornando-a mais fiel ao domínio representado e tornando seus comprometimentos ontológicos explícitos.

Fidedignidade à realidade e clareza conceitual são atributos de qualidade essenciais para modelos conceituais, em geral, e para ontologias de domínio em particular, e são diretamente responsáveis pela efetividade desses modelos como arcabouços de referência para as tarefas de reúso e interoperabilidade semântica como exemplificado em [1-4].

REFERÊNCIAS

[1] Guizzardi, G. Ontological Foundations for Structural Conceptual Models, Telematica Insituut Fundamental Research Series no. 15, The Netherlands, 2005, ISBN 90-75176-81-3.
 [2] Guizzardi, G., The Role of Foundational Ontology for Conceptual Modeling and Domain Ontology Representation, Keynote Paper, Proceedings of 7th International Baltic Conference on Databases and Information Systems (DB&IS), Vilnius, IEEE Press, 2006.

[3] Guizzardi, R.S.S. Agent-Oriented Constructivist Knowledge Management, PhD Thesis, University of Twente, The Netherlands, 2006, ISBN 90-365-2313-3.
 [4] Guizzardi, G., Wagner, G. Some Applications of a Unified Foundational Ontology in Business Modeling, In Ontologies and Business Systems Analysis, M. Rosemann and P. Green (Eds.), IDEA Publisher, 2005.
 [5] Falbo, R.A., ODE: Ontology-based Software Development Environment, IX Congreso Argentino de Ciencias de la Computación, La Plata, Argentina, Outubro 2003.
 [6] Gärdenfors, P., Conceptual Spaces: the Geometry of Thought, USA, MIT Press, 2000.
 [7] Dockhorn, P., Guizzardi, G. et al., Situations in Conceptual Modeling of Context, Proceedings of the 2nd Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE), Hong Kong, IEEE Digital Library, 2006.
 [8] Herre, H. et al., General Formal Ontology (GFO): A Foundational Ontology Integrating Objects and Processes. Technical Report n.8. University of Leipzig, 2006.
 [9] Allen, J.F., Maintaining Knowledge About Temporal Intervals, Communications of the ACM, Vol.26, no. 11, 1983.
 [10] Bottazzi, E.; Ferrario, E., Towards a DOLCE Ontology of Organizations, Journal of Business Process Integration and Management (IBPIM), Inderscience Publisher, 2008 (forthcoming).
 [11] Searle, J., Mind, Language and Society, Basic Books, 2000.
 [12] Conte, R., Castelfranchi, C., Cognitive and Social Action, UCL Press, London, 1995.
 [13] Falbo, R.; Bertollo, G., Establishing a Common Vocabulary for Helping Organizations to Understand Software Processes, Proceedings of the 1st Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE), Enschede, The Netherlands, ISSN 1574-0846, 2005.



Giancarlo Guizzardi nasceu em Vitória, Espírito Santo, Brasil em 22 de abril de 1975. Possui Doutorado em Ciência da Computação (com a mais alta distinção) pela Universidade de Twente na Holanda. Dr. Guizzardi foi pesquisador associado do Laboratório de Ontologia Aplicada (LOA), Instituto de Ciência e Tecnologia da Cognição (ISTC) em Trento, Itália. Atualmente é professor adjunto do Departamento de Informática da Universidade Federal do Espírito Santo (UFES) e membro do Núcleo de Estudos em Modelos e Ontologias (NEMO). Entre seus campos de interesse estão Ontologia Formal e suas aplicações, Engenharia de Ontologias, e Modelagem Conceitual.



Ricardo A. Falbo nasceu no Rio de Janeiro, Brasil em 21 de setembro de 1964. Possui Doutorado em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (COPPE). Dr. Falbo é professor associado do Departamento de Informática da Universidade Federal do Espírito Santo (UFES), membro do Núcleo de Estudos de Modelos e Ontologias (NEMO) e coordenador do Laboratório de Engenharia de Software (LabES). Entre seus campos de interesse estão Engenharia de Ontologias e aplicação de ontologias em Engenharia de Software (Qualidade de Software, Processo de Software, etc).



Renata S.S. Guizzardi nasceu em Volta Redonda, Rio de Janeiro, Brasil em 19 de novembro de 1974. Possui Doutorado em Ciência da Computação pela Universidade de Twente na Holanda. Dra. Guizzardi foi pesquisadora da Divisão de Sistemas de Raciocínio Automatizado (SGA) da Fondazione Bruno Kessler (FBK) em Trento, Itália. Atualmente é pesquisadora DCR da FAPES, colaboradora de pesquisa do Departamento de Informática da Universidade Federal do Espírito Santo (UFES) e membro do Núcleo de Estudos em Modelos e Ontologias (NEMO). Entre seus campos de interesse estão Modelagem Organizacional, Engenharia de Sistemas Multi-Agentes, e Gestão do Conhecimento.