From the Essence of an Enterprise towards Enterprise Supporting Information Systems

Tanja Poletaeva[1], Habib Abdulrab[1]

[1] INSA de Rouen, LITIS lab., Rouen, France
ta.poletaeva@gmail.com, abdulrab@insa-rouen.fr

Abstract. Our research aims at developing a method to link collaborative enterprises with their supporting information systems. The proposed method is founded on the ontology-based enterprise conceptual modeling, the subsequent transformation of enterprise ontology conceptual patterns into factual knowledge of (artificial) cognitive agents, as well as the use of this knowledge on execution of action rules assigned to agents. The discussed ontology conceptual patterns allow better link the metadata of both the production and the intersubjective world of an enterprise and, consequently, to provide support to cognitive agents in their perception of observed situations. Better ñunderstandingò of situations contributes to more accurate run-time execution of agentôs action rules supporting business transactions. The proposed method is demonstrated and discussed in application to a fictitious but realistic case study from the pizza production domain.

Keywords: Foundational Ontology, Enterprise Ontology, Ontology Pattern, Ontology Pattern Reuse, Metadata, Cognitive Agent.

1 Introduction

To effectively respond market demands, members of collaborative enterprise networks or autonomous departments of a networking organization may need virtual collaborative assistants able to (semi-)automate their interactions and information exchange. In addition to being able to follow the formalized patterns of business interactions and actions [1], such virtual assistants are required a number of developed perceptual and cognitive functions for a proper support of the dynamics of business interactions *in connection with* the observed state of affairs, as well as for sharing of information resources. However, developing enterprise information systems able to assign meaning to the observed states of their environment or themselves is still a huge challenge [2]. Henceforth, we use term *cognitive agent* [3-4] when referring to an enterprise information system able to interpret states in an enterprise world and to react to them in accordance with the fixed behavioural patterns and internal representations of the states to be reached.

Our analysis of existing solutions providing the functions similar to those of cognitive agents was made on the assumption that (i) the implemented metadata of a cogni-

adfa, p. 1, 2011.

tive agent underlie its perceptual ability; moreover, (ii) the semantic power[1] of metadata defines agentôs interpretation ability, i.e., an ability to extract the information behind a piece of data. In addition, we believe that the interpretation ability of cognitive agents can be reinforced by the use of codified (implemented) ontologies as semantic domains [5] for the definition of formal and real-world semantics of metadata schemes. Thus, we selected some recently reported works on enterprise supporting information systems, which contain metadata description and refer to the underlying ontologies [6-11]. In these works we assessed the semantics of metadata, a method of its domain specification, a possible representation (by metadata elements) of commitments and claims [1], [12] created in business interactions, the relation between social commitments and a service (or product) lifecycle; and the power of implementation technologies.

In our work, we make an attempt to overcome the drawbacks of existing approaches to data modeling. We elaborate on the abilities of cognitive agents ensured by their metadata, which (i) is built upon the enterprise ontology with formal semantics; (ii) inherits imperative enterprise concepts from the enterprise ontology; (iii) relates a service lifecycle to social commitments; (iv) keeps personal responsibilities of enterprise actors for changes in the production world; (v) creates the possibility for integration of domain knowledge with business interactions.

The semantic domain of the proposed metadata is established by the Formal Enterprise Ontology (FEO), which is derived from a synthesis of (1) the Unified Foundational Ontology (UFO) [5] and (2) the enterprise ontology DEMO (Design and Engineering Methodology for Organizations) [1]. Though we do not provide the details of FEO in this paper, we present some enterprise ontology conceptual patterns [13] built upon this ontology. These patterns have a very high-level of abstraction, they are easy to share, comprehend, re-use and extend [14]. Being codified by means of Semantic Web technologies, these patterns provide strongly interrelated metadata elements, which preserve the formal semantics of FEO.

In this paper, we also propose an approach to data and metadata processing by cognitive agents. Firstly, this approach focuses on the composition of a factual knowledge [15-16] from data and metadata available for agents. Then, we elaborate on the ability of cognitive agents to exploit their factual knowledge on execution of action rules assigned to them. The proposed approach is based on a meta model for modeling organizations proposed in [1] and complement existing solutions aimed at maintenance and control of business interactions in an enterprise (e.g., [7], [17-20]).

The outline of this paper is organized as follows. First, the theoretical background and the methodology of our work are summarized in Sect. 2 and 3 respectively. In Sect. 4, we propose a simplified version of the ontological framework for knowledge management operated by cognitive agents. Then Sect. 5 illustrates the application of

[1] In computer science, the notion of semantics is twofold. In AI research, semantics is understood as a mapping (interpretation) from the language vocabulary to concepts that stand for entities in the real world. In other works, the term (mathematical) ñsemanticsò is used to denote rules for automated interpretation, which are not necessarily related to a real-world conceptualization [5]. In our work, both notions of semantics are used.

the framework to Pizzeria as an example organization. Finally, Sect. 6 presents the final considerations and directions for further research.

2 Theoretical Background

In this section, we briefly present the underlying ontologies of the proposed metadata as well as elaborate on the notion of factual knowledge.

2.1 The Unified Foundational Ontology (UFO)

The philosophically, linguistically and cognitively well-founded foundational ontology UFO was first proposed by Guizzardi in [5] and has been developed in many works afterwards. UFO consists of four main parts: an ontology of endurants (objects, continuants) ï UFO-A [5], an ontology of perdurants (events, occurrents) ï UFO-B [21], an ontology of social entities ï UFO-C [22], and an ontology of services ï UFO-S [23]. Hereafter, we briefly summarize the formal and ontological meta-properties of some types elaborated in the first three parts of UFO. These types are represented by stereotypes in the metamodel of the OntoUML conceptual modeling language [5].

UFO-A explains a number of distinctions among object types. Whilst all types carry a principle of application, only sortal types either provide or carry a uniform principle of identity for their instances. In this research, we exploit the following sortal types: *Kind*, *Subkind*, *Role*, *Phase*. While Kinds provide a principle of identity for their instances, Subkinds carry the principle of identity supplied by Kinds. Moreover, Kinds and Subkinds carry out a meta-property of rigidity being necessarily applied to their instances in every possible world. In contrast, anti-rigidity characterizes a type whose instance(s) can cease to be an instance of that type without ceasing to exist and without altering its identity. For example, a particular individual, which is an instance of type *Student* in one world, can cease to instantiate this type in another world without ceasing to exist as the same individual of type *Person* [5]. Thus, Roles and Phases are distinguished as anti-rigid sortals. A Phase is a relationally independent type whose instantiation is characterized by a change of an intrinsic property of an individual. A Role is a relationally dependent type whose instantiation is obligatory related to other entities.

Non-sortals represent an abstraction of properties that are common to multiple disjoint kinds and, therefore, do not carry a unique principle of identity for their instances. *Category* represents a rigid and relationally-independent non-sortal type that aggregates essential properties common to different kinds. *Role Mixin*, in turn, represents an anti-rigid and relationally dependent non-sortal type that aggregates properties common to different roles.

Another important distinction in the UFO ontology is within the categories of relations. It recognizes two broad categories of relations, namely, *formal relations* and *material relations*. Formal relations hold between two or more entities because of very nature of these entities, without any further intervening individual. Conversely, material relations have a material structure of their own, which mediates the connected entities and inheres in the mereological sum of them [24]. Such mediating entities constitute the extension of *Relators*. For example, a *medical treatment* connects a patient with a medical unit; a *marriage* connects a wife and a husband [5].

Axiomatization of s*ituations*, *events*, and *dispositions* was summarized in UFO-B. Situations are special type of endurants. They are complex entities that are constituted by possibly many endurants (including other situations). Situations can be factual or counterfactual. Factual situations (or Facts) are said to obtain at particular time points. Properties that are only manifested in particular situations on the occurrence of certain triggering events are called *dispositions*. Dispositions are manifested through the occurrence of resulting events and state changes [25].

UFO-C incorporates *intentionality* to the basic core provided by UFO-A and UFO-B. In this context, UFO distinguishes between Agentive and Non-agentive substantial individuals, termed *agents* and *objects*, respectively. As opposed to objects, agents are capable of bearing special kind of intrinsic properties named *intentional moments*. Intentionality of agents should be understood as the capacity of their properties to refer to possible situations of reality. Every intentional moment has a type (e.g., *belief*, *desire*, *intention*) and a propositional content represented by a *proposition*. The latter being an abstract representation of a class of situations referred by that intentional moment.

Intentions are desired state of affairs for which the agent commits at pursuing (an intention is an internal commitment). For this reason, intentions cause the agent to perform *actions*. Actions are intentional events, i.e., events with the specific purpose of satisfying the propositional content of some intention of an agent.

Communicative acts (special kinds of actions) can be used to create *social moments* (*commitments* and *claims*). Thus, social moments are types of intentional moments that are created by the exchange of communicative acts between parties and the consequences of these exchanges. In this view, language not only represents the reality but also creates a part of reality. The later ontological claim tightly correlates with the LAP foundations of DEMO (ref. section 2.2).

2.2 The DEMO Theory and Methodology of Enterprise Ontology

In this work, we employ the OntoUML [5] conceptual modeling language to extend real-world semantics of the modeling constructs of the DEMO Theory and Methodology of Enterprise Ontology [1]. Based on the strong theoretical basis, the DEMO methodology facilitates creation of ontological conceptual models that are essential, complete, free from logical contradictions, compact and succinct, independent of their realization and implementation issues [1]. The theory that underlies the DEMO meta-models is called the PSI-theory (the Performance in Social Interaction) [1]. Hereafter, we briefly summarized the axioms of this theory.

The Operation Axiom states that an enterprise is as a social system, of which the elements are social individuals or *actors*, capable to negotiate by performing *coordination acts* (C-acts) and to contribute to bringing about the goods or services by performing *production acts* (P-acts). By performing coordination acts, actors express their *intensions* and comply with *commitments* towards each other regarding the performance of production acts [1]. By performing both kinds of acts, actors transfer the world into the new states characterized by resulted *coordination facts* (C-facts) and (if any) *production facts* (P-facts). Granted authority to perform particular acts in a responsible way, a social individual of an enterprise fulfills an *actor role* [1]. In his fulfillment of an actor role, a social individual becomes an actor.

The Transaction Axiom states that coordination acts and production acts occur according to the uniform communication patterns. Each pattern, also called a *transaction pattern*, always involves two *actor roles* (the initiator and the executor) and consists of certain types of coordination acts related to a particular type of production acts. *Transaction* is a sequence of acts that is a path through the complete transaction pattern [1].

The Composition Axiom states that according to the way of initiation every transaction is caused in some other transaction, or is a customer transaction requested by an actor in the environment, or is a self-activation transaction requested by an internal actor to itself.

The Distinction Axiom states that there are three distinct human abilities playing a role in the operation of actors, called performa, informa and forma. The performa ability concerns the ability of human beings to produce new original P-facts while directly or indirectly communicating. The informa ability regards the intellectual capacity of human beings, the ability to reason and to interpret the information provided. The forma ability concerns the ability to handle data independently of its meaning. While only humans can create original facts that cannot be derived from existing facts, cognitive agents may facilitate the informa and forma abilities.

2.3 Factual Knowledge of Cognitive Agents

The notion of fact is often overloaded. As Russell noted in [26], ñwhat is complex in the world is a factò. In this section, we elaborate on the notion of fact and its application to knowledge representation of cognitive agents.

Wittgenstein defined facts separately from things and independent from each other: ñ1.13 The facts in logical space are the worldé 1.21 Each item can be the case or not the case while everything else remains the sameò [26]. Being different from things, facts represent their states: ñ2 What is the case ï a fact ï is the existence of states of affairsé 2.01 A state of affairs (a state of things) is a combination of objects (things)ò [26].

According to Wittgenstein [26], at the most fundamental level there are only objects, and no properties or relations. This idea was inherited and elaborated by Dietz in [15]. Thus, Dietz extended the notion of facts by considering them as instantiations of types, where a type (or fact type) was defined as a prescription of the collection of properties that our mind applies when we perceive concrete things.

In opposite to Wittgenstein, Russell explained the notion of facts through properties and relations: ñWe express a fact, for example, when we say that a certain thing has a certain property, or that it has a certain relation to another thing; but the thing which has the property or the relation is not what I call a factò [27]. By using the notion of moment[2] from UFO [5], Russellôs definition of facts can be rephrased as: *a fact is a conceptual property referring to (i) the existence of an endurant and a moment (intrinsic or relational) that inheres in or existentially dependent on this endurant; or (ii) the existence of two formally related endurants.*

Our work is founded on the ontological categories and distinctions put forth by the UFO. Thus, we adopted the latter definition of facts, but with the following extension derived from [15]: *a conceptual property referring to a particular instantiation of a type is also considered as a fact*.

According to the adopted definition of facts, each elementary fact can be presented in the form binary predicate of a first-order language. For instance, the fact <myPizza is of type PizzaKindX> can be expressed by the predicate: *instantiates(myPizza, PizzaKindX)*. In case such elementary facts are implemented as RDF statements or as predicates of DL-based versions of OWL, they can be integrated and subject to automated reasoning.

Interacting cognitive agents constitute the enterprise supporting information system. At any moment a state of this system is simply defined as a set of states of its constituting agents. What is the state of an agent? Following Dietz [1], we assume that cognitive agents may have an ability to partially fulfill cognitive abilities of human actors. Thus, a particular state of an agent is defined by the set of facts that can be inferred by this agent from its data and metadata, i.e., by a factual knowledge of an agent. It is important to emphasize that in opposite to Wittgensteinôs or Russellôs fact, facts inferred by cognitive agents relate to data elements, and not to real world entities. A state change (a transition) of an agent is a succession of qualitatively dissimilar existentially-dependent states.

3 Research Methodology

This research is carried out by making use of the Design Science Research methodology [28]. The following stages were performed so far:

Stage 1. The identity of a problem in the field of Enterprise Engineering using a simplified case study. At this stage, the required cognitive and collaborative abilities of virtual collaborative assistants were specified.

Stage 2. A literature review, which confirmed that no full solution exists. We formulated two classes of problematic contexts. The first one is the class of solutions providing the description of ontology-based metadata and its use in the automated execution of action rules. The second one is the class of solutions related to the formalization and automated execution of action rules, as well as the solutions aimed at automated control and support of business interactions.

Stage 3. The generative mechanisms on which our solution is grounded are the Unified Foundational Ontology (UFO), the DEMO Enterprise Ontology, and the theories of factual knowledge.

The following stages should be completed:

---

[2] ñA moment is an endurant that inheres in, and, therefore, is existentially dependent of, another endurantò [5].

Stage 4. Designing and implementing a software prototype of cognitive agent as an extension of a multi-agent platform. This stage meets the requirements of the Generic System Development Process (GSDP) formulated by Dietz in [1].

Stage 5. Validation of the proposed solution through simulations and case studies. Then, dissemination the contribution in the interested community, using technical reports, presentations and publications. Learning from the interactions and improvement of the solution starting from Stage 1.

4 Ontological Foundations of Cognitive Agents

In this section we describe the essential parts of the proposed enterprise knowledge management framework. Due to lack of space, we consider only two enterprise ontology patterns (Sect. 4.1) and only one type of transition rules (Sect. 4.2). Other formal enterprise ontology patterns were discussed in [13], while the full version of the formal enterprise ontology created upon the synthesis of UFO and DEMO, has not been published yet. In Sect. 4.3, we demonstrate how factual knowledge is adopted by cognitive agents when executing their action rules. In order to apply this framework, all parts have to be specified (and extended) for a particular case study.

4.1 Enterprise Ontology Patterns

Enterprises are social systems whose elements (or social individuals) are human beings able to enter and comply with commitments [1]. Granted authority to perform particular acts in a responsible way, a social individual of an enterprise fulfills an *actor role* [1]. In his fulfillment of an actor role, a social individual becomes an actor. Obviously, a social individual may cease to be an actor without ceasing to exist. Moreover, an actor role can be instantiated by agents of different kinds, e.g. persons, organizations or organizational units. Thereby, *Actor Role* is a *Role Mixin* stereotyped by <<roleMixin>> in OntoUML model depicted in Fig. 1.

The notion of transaction discussed in DEMO [1] is articulated in the UFO categories as follows. At each time the business relation between two actors holds, the *transaction* is constituted by a mereological sum of all mutual commitments made by actors in their negotiation about a particular change in the production world (i.e., a production result). Thus, each transaction is a *relator* mediating two actors (Fig. 1).

In the broader sense, transactions are mereological sums of all their constituent relational qualities [24] inhering in one interacting actor and directly or indirectly existentially dependent on another one. These qualities and their changes can be specified in conceptual models by attributes of transactions.

Each coordination act performed by an actor towards his addressee contains an intention and a proposition [1]. With the *intention*, an actor proclaims his ósocial attitudeô with respect to the *proposition*. The standard transaction may contain coordination acts with the following intentions: request, promise, state, accept, decline, quit, reject, and stop [1]. With the proposition, an actor proclaims an abstract representation of a class of desired situations [1].

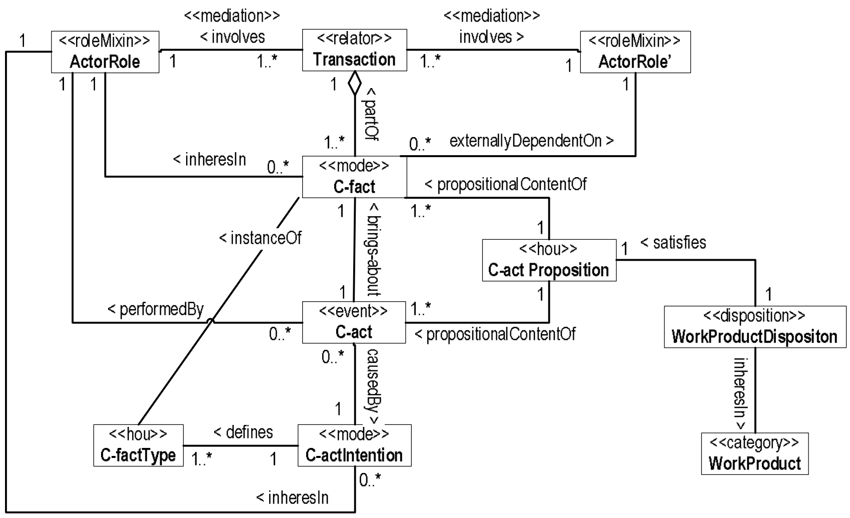

Fig. 1. The pattern of Coordination Acts and Facts

In accordance with many standards, the portion of reality which is subject to changes in business processes can be abstracted by the category *Work Product*. Thus the foregoing definition of coordination acts was formalized by the Coordination Act pattern comprised by the types: *Actor Role, C-act, C-act Intention, C-act Proposition, Work Product Disposition, Work Product*, as well as their interrelations depicted in Fig. 1.

We propose to model instances of the *Proposition* type as specializations (subtypes) of *Work Product Disposition*, where the extension of the latter is a class of desired situations such as creation or termination of an object, a relation between objects, or a qualitative property of an object. For example, the proposition of C-acts and C-facts appeared in a transaction of type *Membership Registration* can be formulated as ó(new) membership has been startedô or simply, ó*Membership Started*ô, where the latter is a possible specialization of *Work Product Disposition*. Thus, the *Proposition* type is a high-order universal (a powertype), i.e. a rigid sortal whose instances are types [29]. Following [30], we extended the OntoUML metamodel by introducing the stereotype <<hou>> to represent high-order universals.

A coordination act *brings about* a situation, which *triggers* a (social) *commitment* of one agent towards another regarding the proclaimed intention and the proposition. This commitment is a *coordination fact* (*C-fact*) [1]. Examples of coordination facts are: ñ*membership has been started* is requested asapò, ñ*membership has been started* is promised asapò, where ñ*membership has been started*ò is the propositional content of both C-acts and their resulted C-facts, and ñmembershipò is an identifiable instance of kind *Membership*, where *Membership* is a specialization of the category *Work Product*. For the sake of simplicity, in this version of patterns, we omit from consideration the time part of propositions.

As a social commitment, a *C-fact inheres* in one of the negotiating actors performing a C-act, and is *externally dependent* on the target actor. Hereby we consider a C-fact being a relational quality (a *mode*) that contributes to constitute the relationship between two actors, i.e., each *C-fact* is always a *part of* some *transaction*. Moreover, we assume that each C-fact inherits the (fact part of) C-act proposition from its activating C-act.

C-acts with particular intention define a partition of the generalization set of C-fact types (*C-factType* powertype). In other words, all C-facts brought about by C-acts with particular intention, are instances of particular specialization of the *C-fact* type. For instance, *Work Product Disposition Requested* comprises instances of *C-fact* that result C-acts with the intention ‘request’, or *Work Product Disposition Promised* comprises instances resulted C-acts with the intention ‘promise’.

The foregoing definition of coordination facts was formalized by the Coordination Fact pattern. This pattern comprises the types: *Actor Role*, *Transaction*, *C-fact*, *C-fact Type*, *C-act Proposition*, *Work Product Disposition*, *Work Product* as well as their interrelations depicted in Fig. 1.

Given patterns of Coordination Acts and Facts can be implemented in OWL and SWRL by means of the OntoUML Lightweight Editor[3]. Depending on the required semantics of metadata, one can choose different methods of implementation. The implementation scenario where instances are represented in a reified way according to the OntoUML stereotypes, preserves ontological aspects, such as concept hierarchy. In other words, this scenario keeps the OntoUML stereotypes in metadata. However, the further discussion of implementation methods and issues is out of the scope of this paper. Independently of the implementation method, the types depicted in Fig. 1 are represented by classes or collections (owl:class, owl:unionOf), and the depicted associations are represented by object properties (owl:ObjectProperty).

4.2 Transition Rules of an Enterprise

Interacting human actors and passive objects in their environment constitute the enterprise world. A state of this world at every point in time is the set of relevant coordination and production facts. A coordination act can potentially trigger a causal law. Hereafter we specify one type of transition rules representing state transitions of an organization caused by a coordination act (though based on the CRISP model of DEMO [1], given type of transition rules has meaningful differences).

Let *CF* be the union of the extensions of the C-fact type of which *instances an actor role* is prescribed to deal with, and *S* be the union of all possible facts (including instances of C-and P-fact types) that an actor may need to know for reacting C-facts. Let *I* be the set of intentions that may be evoked in the actor, let *P* be the set of propositions which intentions are about, and *CA* be the union of all possible C-acts with the specified intention and proposition.

---

3 OntoUML Lightweight Editor: https://github.com/nemo-ufes/ontouml-lightweight-editor

A set of transition rules is a total function TR: *(CF, S) → (CA, I, P)* that maps each C-factum of *CF* appeared in a particular state *S* to C-acts from *CA* with the related instances of intention from *I* and proposition from *P*.

Due to formalization of a social system, transition rules are formulated for an actor role as the kernel of local state changes. Moreover, transition rules are formulated in assumption that actors in an enterprise commit to respond to particular types of C-facts according to the roles they play. In addition, the proposed type of transition rules has to be specified according to the standard transaction pattern from [1]. All these conditions stipulate rather deterministic enterprise models, while it is the actorôs decision whether or not to act, and how to act.

4.3 As Factum Is Fictum, and Making Is Mockingé

In this section we discuss factual and procedural knowledge of cognitive agents. Following the definition of facts given in Section 2.3 and the metadata resulted the implementation of the patterns from Section 4.1, we can define the core sets of (atomic) facts representing coordination acts and facts. When observing (or storing related data) a new C-act or C-fact, cognitive agents are required to verify the set of interrelated facts.

Each instance of coordination act ca is perceived by a cognitive agent together with a number of facts:
instantiates(ca, C-act) ∧ *instantiates(ar, ActorRole)* ∧ *performedBy(ca, ar)* ∧
instantiates(cap, C-actProposition) ∧ *propositionalContentOf(cap, ca)* ∧
causedBy(ca, i) ∧ *instantiates(i, C-actIntention)* ∧ *inheresIn(i, ar)* ∧
satisfies(WorkProductDisposition, cap)

Each instance of coordination fact *cf* is perceived by a cognitive agent together with the following facts:
instantiates(cf, C-fact) ∧ *instantiates(ar, ActorRole)* ∧ *instantiates(arô, ActorRoleô)* ∧
inheresIn(cf, ar) ∧ *externallyDependentOn(cf, arô)* ∧
instantiates(cap, C-actProposition) ∧ *propositionalContentOf(cap, cf)* ∧
satisfies(WorkProductDisposition, cap)

For coordination facts and acts at the result phase of business transactions, state changes (dispositions) of a work product should meet the following facts in addition to ones listed above:
instantiates(wp, WorkProduct) ∧ *instantiates(wpd, WorkProductDisposition)* ∧
inheresIn(wpd, wp)

With the use of these core sets of facts, cognitive agents can infer responsibilities of actors for coordination facts and acts, infer the relation between coordination facts and the type of required changes in the production world, interrelate coordination acts and facts through C-act propositions. For the sake of simplicity, in this version of patterns, we omit from consideration the time part of propositions. While time parts will complement the proposed core sets of facts.

On execution of transition rules of the type specified in Section 4.2, a cognitive agent operates not with the simple instances of coordination facts and acts, but with the core sets of facts specified from the Coordination Acts and Facts pattern. Moreover, an agent infers facts that are necessary to know in order to react to the observed coordination fact. On execution of transition rules, an agent ensures that the coordination acts and facts with the same propositions belong to the same (instance of) transaction.

5 Practical Implications of the Ontological Foundations

In this section, we shortly present a case study, applying the enterprise knowledge management framework to control of pizzas order. One can imagine a simple Pizzeria where only pizzas are made on receiving a new order. The owner of the company decided to implement a Web Service in order to provide the possibility of pizzas order through the Internet. New orders should be not only stored in the database, but thoroughly processed by a cognitive agent. Then the agent should provide the adequate reaction to customer needs.

At the first stage, the conceptual model of the Order Completion transaction was created (Fig. 2), where *Customer* is the initiator and *Order Manager* is the executor, *Purchased Order* specifies *Work Product*, *Purchased Order Requested* is a type of coordination facts, *Purchased Order Completed* is a type of *Work Product Dispositions.*

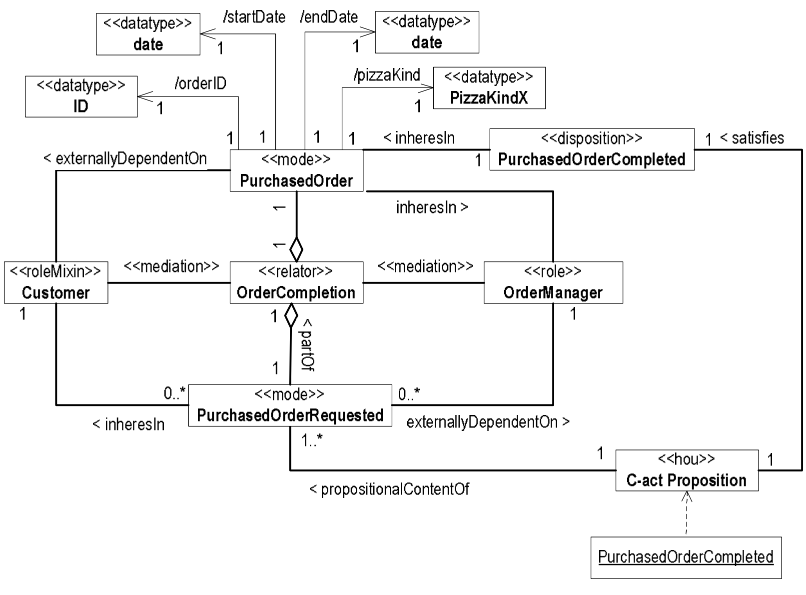

Fig. 2. The conceptual model of the OrderCompletion transaction type with the specified coordination fact type PurchasedOrderRequested

We suppose that two facts are verified by a cognitive agent on observing a new pizza requested (a coordination fact). For instance, it is required that ñthe recipe of the ordered pizza kind is availableò. This fact can be expressed by the triple: ñPizza-

KindX has recipe PizzaKindXRecipeò. Formally, *recipe(PizzaKindX, PizzaKindXRecipe)* [C1]

Similarly, the production fact ñthe price of the pizza kind X was assignedò which is obtained in a ñprice assignmentò transaction, is formally presented by the expression, numbered [C2]:

instantiates(PricedPizzaKindX, C-actProposition) ∧ *inhereIn(PricedPizzaKindX, PizzaKindX)* ∧ *propositionalContentOf(PricedPizzaKindX, PricedPizzaKindXStated)* ∧ *propositionalContentOf(PricedPizzaKindX, PricedPizzaKindXAccepted)* ∧ *price(PizzaKindX, PriceValue)*

At the next step, defined sets of facts allow modeling the elements of transition rules. To exemplify the specification of transition rules, we formulate one transition rule for the actor role óOrderManagerô.

On observation of a request for a new purchase (C-fact of type *WorkProductDispositionRequested*), an óOrderManagerô verifies (1) the existence of recipe for the requested PizzaKindX [C1]; and (2) the price availability for a chosen pizza kind [C2]. If both conditions are satisfied, the óOrderManagerô has the intention to promise the order completion. For this rule, the proposition of both the observed C-fact and the triggered C-act is the same. This proposition is about the existence of two dispositions: *PizzaKindXBaked* and *OrderPaidState*. Formally,

∀*cf instantiates(cf, PurchasedOrderRequested)* ∧
propositionalContentOf(PizzaKindXBaked, cf) ∧
propositionalContentOf(OrderPaidState, cf) ∧
inhereIn(PizzaKindXBaked, PizzaKindX) ∧ ∃*x (x::C2)* →
(∃*promise (promise::C-actIntention* ∧
propositionalContentOf(PizzaKindXBaked, promise) ∧
propositionalContentOf(OrderPaidState, promise))

6 Final Considerations

In this work, we discussed our vision of the following important components of enterprise supporting information systems: (i) the ontological core of metadata; and (ii) the method of metadata processing.

Our method of data modeling provides the basis for semantically reach information footprints of business interactions in data models, which make lifecycles of services (or products) traceable and controllable through personal commitments. Factual knowledge of agents and the sets of facts defined for a comprehensive representation of ontological entities, ensures proper semantics of data elements.

Based on the proper ñunderstandingò of a situation, enterprise information systems should be able to decide on the consequent collaborative action. Thus, this work would hopefully contribute to nowadays attempts to make process management a matter for computers.

At the next stage of our research, we finalize codification of enterprise ontology patterns, design and implement cognitive agents, implement reasoning algorithms for factual knowledge processing, and, finally, define the method to integrate reasoning algorithms and execution of business rules.

References

1. Dietz, J.L.G.: Enterprise Ontology ï Theory and Methodology. Springer, Heidelberg (2006)
2. Panetto, H., Zdravkovic, M., Jardim-Goncalves, R., Romero, D., Cecil, J., Mezgar, I.: New perspectives for the future interoperable enterprise systems. Computers in Industry, Special Issue ñFuture Perspectives on Next Generation Enterprise Information Systems: Emerging Domains and Application Environmentsò, pp. In Press. Elsevier (2016)
3. Castelfranchi, C.: Guarantees for Autonomy in Cognitive Agent Architecture. In: Wooldridge, M.J., Jennings, N.R. (eds.) Workshop on Agent Theories, Architectures, and Languages, ECAI-94. LNCS, vol. 890, pp. 56-70. Springer, Heidelberg (1995)
4. Guizzardi, G., Wagner, G.: Towards an Ontological Foundation Of Agent-Based Simulation. In: Proceedings of the 2011 Winter Simulation Conference (WSC), pp. 284-295. IEEE (2011)
5. Guizzardi, G.: Ontological foundations for structural conceptual models. Telematics Instituut Fundamental Research Series, No. 015, The Netherlands (2005). ISSN 1388-1795
6. Krouwel, M.R., Opât Land, M.: Combining DEMO and Normalized Systems for Developing Agile Enterprise Information Systems. In: Albani, A., Dietz, J.L.G., Verelst, J. (eds.) First Enterprise Engineering Working Conference, EEWC 2011. LNBIP, vol. 79, pp. 31-45. Springer, Heidelberg (2011)
7. Dudok, E., Guerreiro, S., Babkin, E., Pergl, R.., van Kervel, S.J.H.: Enterprise Operational Analysis Using DEMO and the Enterprise Operating System. In: Aveiro, D., Pergl, R., Valenta, M. (eds.) 5th Enterprise Engineering Working Conference, EEWC 2015. LNBIP, vol. 211, pp. 3-18. Springer, Switzerland (2015)
8. de Jong, J.: Designing the Information Organization from Ontological Perspective. In: Albani, A., Dietz, J.L.G., Verelst, J. (eds.) First Enterprise Engineering Working Conference, EEWC 2011. LNBIP, vol. 79, pp. 1-15. Springer, Heidelberg (2011)
9. Baclawski, K., Chan, E., Dragan, L., Durusau, P., Lee, D., Yim, P., Yin, Y.: ICOM: A Framework for Integrated Collaborative Work Environments. In: Haller, A., Huang, G., Huang, Z., Paik, H., Sheng, Q.Z. (eds.) Web Information Systems Engineering ï WISE 2011 and 2012 Workshops. LNCS, vol. 7652, pp. 147-158. Springer, Heidelberg (2013)
10. Opdahl, A.L., Berio, G., Harzallah, M., Matulevicius, R.: An ontology for enterprise and information systems modelling. Applied Ontology, vol. 7, pp. 49-92. IOS Press (2012)
11. Han, K.H., Park, J.W.: Process-centered knowledge model and enterprise ontology for the development of knowledge management system. Expert Systems with Applications, vol. 36, issue 4, pp. 7441-7447. Elsevier Ltd. (2009)
12. Nardi, J. C., De Almeida Falbo, R., Almeida, J.P.A., Guizzardi, G., Ferreira Pires, L., Van Sinderen, M. J., Guarino, N., Fonseca, C.M.: A Commitment-based Reference Ontology for Services. Information Systems, vol. 54, pp. 263-288. Elsevier Ltd. (2015)
13. Poletaeva, T., Abdulrab, H., Babkin, E.: From the Essence of an Enterprise towards Enterprise Ontology Patterns. In: Aveiro, D., Pergl, R., Gouveia, D. (eds.) Advances in Enterprise Engineering X (EEWC 2016). LNBIP, vol. 252, pp. In Press. Springer, Switzerland (2016)

14. Ruy, F.B., Reginato, C.C., Santos, V.A., Falbo, R.A., Guizzardi, G.: Ontology Engineering by Combining Ontology Patterns. In: Johannesson, P., Lee, M.-L., Liddle, S.W., Opdahl, A.L., Lopez, O.P. (eds.) ER 2015. LNCS, vol. 9381, pp. 173-186. Springer (2015)
15. Dietz, J.L.G.: The FI theory ï understanding information and factual knowledge. The TEE series (Theories in Enterprise Engineering), TEE-1, v2.3 (2015) [online: http://www.ciaonetwork.org/uploads/eewc2015/ee_theories/theories/]
16. Diaconescu, I.M., Wagner, G.: Agent-Based Simulation with Beliefs and SPARQL-Based Ask-Reply Communication. In: International Workshop, MABS 2009. LNCS, vol. 5683, pp. 86-97. Springer, Heidelberg (2010)
17. Guerreiro, S., Vasconcelos, A., Tribolet, J.: Enterprise Dynamic Systems Control Enforcement of Run-Time Business Transactions. In: Second Enterprise Engineering Working Conference, EEWC 2012. LNBIP, vol. 110, pp. 46-60. Springer, Heidelberg (2012)
18. Barjis, J., Chong, S., Dietz, J. L. G., Liu, K.: Development of Agent-based E-commerce Systems using the semiotic approach and the DEMO transaction concept. International Journal of Information Technology & Decision Making, vol. 1, no. 3 (2002)
19. Hunka, F., Zacek, J.: A new view of REA state machine. Applied Ontology, vol. 10, pp. 25-39. IOS Press (2015)
20. Figueira, C., Aveiro, D. : A New Action Rule Syntax for DEmo MOdels Based Automatic worKflow procEss geneRation (DEMOBAKER). In: 4th Enterprise Engineering Working Conference, EEWC 2014. LNBIP, vol. 174, pp. 46-60. Springer, Switzerland (2014)
21. Guizzardi, G., Wagner, G., De Almeida Falbo, R., Guizzardi, R.S.S., Almeida, J.P.A.: Towards Ontological Foundations for the Conceptual Modeling of Events. In: 32th International Conference ER 2013. LNCS, vol. 8217, 327-341. Springer, Heidelberg (2013)
22. Guizzardi, G., Falbo, R.A., Guizzardi, R.S.S.: Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology. In: XI Iberoamerican Workshop on Requirements Engineering and Software Environments, pp. 244ï251 (2008)
23. Nardi, J. C., De Almeida Falbo, R., Almeida, J.P.A., Guizzardi, G., Ferreira Pires, L., Van Sinderen, M. J., Guarino, N., Fonseca, C.M.: A Commitment-based Reference Ontology for Services. Information Systems, vol. 54, pp. 263-288. Elsevier Ltd. (2015)
24. Guarino, N., Guizzardi, G.: ñWe Need to Discuss the Relationshipò: Revisiting Relationships as Modeling Constructs. In: 27th International Conference, CAiSE 2015. LNCS, vol. 9097, pp. 279-294. Springer, Heidelberg (2015)
25. Guizzardi, G., Wagner, G.: Dispositions and Causal Laws as the Ontological Foundation of Transition Rules in Simulation Models. In: 2013 Winter Simulation Conference, pp. 1335-1346. IEEE (2013)
26. Wittgenstein, L.: Tractatus logico-philosophicus (German text with an English translation by C.K. Ogden). Routledge & Kegan Paul Ltd., London (1922)
27. Russellôs Logical Atomism, ed. D.Pears. London: Fontana/Collins (1972)
28. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly, vol. 28, no. 1, pp. 75-105 (2004)
29. Guizzardi, G., Almeida, J.P.A., Guarino, N., Carvalho, V.A.: Towards an Ontological Analysis of Powertypes. In: International Workshop on Formal Ontologies for Artificial Intelligence (FOFAI 2015), 24th International Joint Conference on Artificial Intelligence (IJCAI 2015). CEUR-WS, vol. 1517 (2015)
30. Falbo, R.A., Ruy, F.B., Guizzardi, G., Barcellos, M.P., and Almeida, J.P.A.: Towards an Enterprise Ontology Pattern Language. In: 29th Annual ACM Symposium on Applied Computing, pp. 323ï330. ACM (2014)