An Ontology-Based Semantic Foundation for ARIS EPCs

Paulo Sérgio Santos Jr., João Paulo A. Almeida, Giancarlo Guizzardi

Ontology & Conceptual Modeling Research Group (NEMO)
Computer Science Department, Federal University of Espírito Santo (UFES)
Av. Fernando Ferrari, s/n, Vitória, ES, Brazil
paulossjunior@inf.ufes.br; jpalmeida@ieee.org; gguizzardi@acm.org

ABSTRACT

This paper presents an ontological analysis of the EPC (Event-driven Process Chain) business process modeling notation supported in the ARIS Toolset. This ontological analysis provides an interpretation of the modeling elements in EPC diagrams in terms of the UFO foundational ontology. This enables us to define the precise real-world semantics for business process models represented through EPCs and allows us to identify problems affecting the clarity and expressiveness of EPCs. In our analysis, we consider an up-to-date metamodel of the ARIS Method that we have defined in our earlier work, which specifies EPCs as currently supported by the ARIS Toolset. To the best of our knowledge, the interpretation proposed here is presently the most complete ontological account for EPCs.

Keywords

Business Process Modeling, Event-Driven Process Chains (EPCs), ARIS, Ontological Analysis, Foundational Ontologies.

1. INTRODUCTION

The ARIS (***AR**chitecture for integrated **I**nformation **S**ystems*) Method proposes a framework for enterprise modeling whose main goal is to allow the description and development of information systems that are integrated to the organizational structure through business processes [6][26].

The ARIS framework is structured in terms of five different views (organization, data, control, function and output) and three abstraction layers (Requirements Definition, Design Specification and Implementation Description)[6][26]. In this paper, we are interested in the *control view* as applied to the *Requirements Definition* layer. In this layer, the *control view* describes information-processing business processes through a set of related functions. Since these functions represent potentially complex organizational tasks, the *control view* is used for modeling business processes and business activities in the ARIS method.

Each view in ARIS has its own diagrammatic language. In the case of the control view, business processes are modeled through diagrams known as *Event-Driven Process Chains* (EPCs)[6][26], which include modeling elements such as events, functions, rules and others.

The importance of EPCs in the practice of business process modeling can be attested by the existence of several successful commercial tools which offer support for these diagrams, e.g., IDS Scheer's ARIS Toolset, Microsoft's Visio and BOC's ADONIS. Further, EPCs have been used in the documentation of the widely-employed SAP R/3 enterprise resource planning system, which has led to the SAP Reference Model with over 600 EPCs [19].

Due to the economic value of EPCs, several academic studies have considered different aspects of the language. For example, [17], [20] and [21] have proposed transformations between EPCs and other (platform-independent or platform-specific) models using Model-Driven Development techniques. In addition, several studies in the literature propose formal execution semantics for EPCs using Petri nets [18][24][34][35], with the purpose of demonstrating the absence of structural problems within the language and its models and to assure that the business processes specified in EPCs are prone to unambiguous execution.

In [5], Davis, Green and Rosemann define the semantics of EPCs through ontological interpretation, i.e., by defining how the elements of an EPC correspond to concepts defined in a reference ontology. Ontological interpretation differs from formal execution semantics definition in that the former describes a language in terms of the "real-world" phenomena it represents, while the latter defines a mapping from the language into an abstract mathematical semantic domain (e.g., "places", "transitions" and "tokens" in the case of Petri nets). We defend that both ontological interpretation and formal execution semantics definition should be applied to a process modeling approach, leading to a comprehensive semantic foundation which clarifies the relation between models and the corresponding modeled phenomena, while also enabling simulation, automated analysis and execution.

This paper focuses on the issue of ontological interpretation for EPCs with the following contributions: **(i)** a rigorous definition of the EPC modeling language in terms of real-world entities in the well-founded UFO upper-level ontology; **(ii)** the identification of inappropriate elements of the EPC language, using a systematic ontology-based analysis approach ([10][11][15][27]); and **(iii)** recommendations for improvements of the EPC language to resolve the issues identified (such as ontological mis-interpretations of the language elements and certain usage problems derived from unsoundness, non-lucidity, non-laconicity and incompleteness [10][15]).

To the best of our knowledge, the interpretation proposed here is the most complete ontological account for EPCs to this date (see section 5 for a detailed discussion on the relation between our approach and the one presented in [5][10]). In addition, previous works (such as [5], [10] and [31]) have relied on the metamodels in Scheer's original proposal[26]. As we have discussed in [25] , the language metamodel in the ARIS Toolset is significantly

SAC'10, March 22-26, 2010, Sierre, Switzerland.

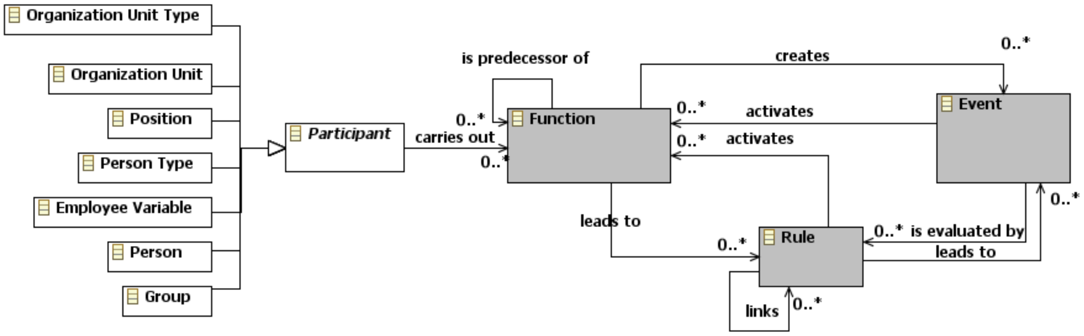

Figure 1 - Fragment of Business Process metamodel of ARIS Method [25].

different from the metamodels in Scheer's original proposal [26]. As a consequence, the approaches based on Scheer's metamodels [26] do not consider the characteristics of the modeling language as actually implemented and employed in enterprises worldwide.

This paper is organized as follows: section 2 presents the EPC metamodel, section 3 introduces the ontological concepts used in our analysis and section 4 provides an interpretation for each metamodel element. Section 5 discusses related work and, finally, section 5 presents our conclusions and discusses future work.

2. BUSINESS PROCESS METAMODEL OF LANGUAGE OF ARIS METHOD

Figure 1 shows a fragment of the metamodel of the business process modeling language EPC used in the ARIS Method. This fragment was excavated in our earlier work by using the approach described in [25] and defines the abstract syntax of the language as currently supported by the ARIS Toolset. The main metaclasses in this fragment are: *Participant*, *Event*, *Rule* and *Function*.

The *Participant* metaclass is an abstract metaclass which subsumes the following organizational elements: *Organization Unit Type*, *Organization Unit*, *Position*, *Person Type*, *Person*, *Group* and *Employee Variable*. These organization elements belong to the ARIS Organizational model and are referenced in an EPC to describe participants in organizational activities. Figure 2 shows an example of an EPC which instantiates the metamodel depicted in Figure 1. In this model, the "*Client*" and "*Seller*" elements are instances of *Position* and *Person Type* metaclasses, respectively.

The *Function* metaclass is a basic element for EPC process modeling. According to the ARIS Toolset online documentation, the element *Function* represents either a technical task or a task performed on some object, with the purpose of achieving one or more business goals [26]. A task can be performed by either a person or an application system [26], and has inputs – such as information or raw material – and outputs, such as new information or products. Furthermore, tasks can consume and create organizational resources during their execution [6]. In Figure 2, "*Request purchase*" and "*Inform client*" are examples of tasks (instances of the *Function* metaclass).

The *carries out* meta-association between the *Participant* and *Function* elements indicates that one or more participants of the business process will be responsible for performing the task. In Figure 2, this meta-association is instantiated and is represented graphically by the positioning of tasks in the lanes that correspond to *Participant* instances ("Client" is responsible for carrying out "Request purchase", while "Seller" is responsible for "Analyze purchase request", "Finish purchase" and "Inform client").

The *is predecessor of* meta-association of the *Function* element is used to indicate that a particular task precedes another task in the business process. For example, we can express that the "Request purchase" task precedes the "Analyze purchase request" task (as depicted in Figure 2). (While both Scheer [26] and Davis [6] defend that every relation between functions in EPCs should be mediated by an event, this restriction is not enforced by the ARIS Toolset. Hence, we include the *is predecessor of* meta-association in our excavated metamodel and discuss it further in our analysis.)

An *Event* represents a state which is relevant to the process management and affects the flow of execution. In other words, *Events* are said to establish the preconditions and postconditions for each stage of the process [6]. Preconditions represent a state of reality which triggers one or more tasks, while postconditions, in their turn, represent a state of reality that exists only after the task has been performed. In ARIS, *Events* may be the result of tasks in

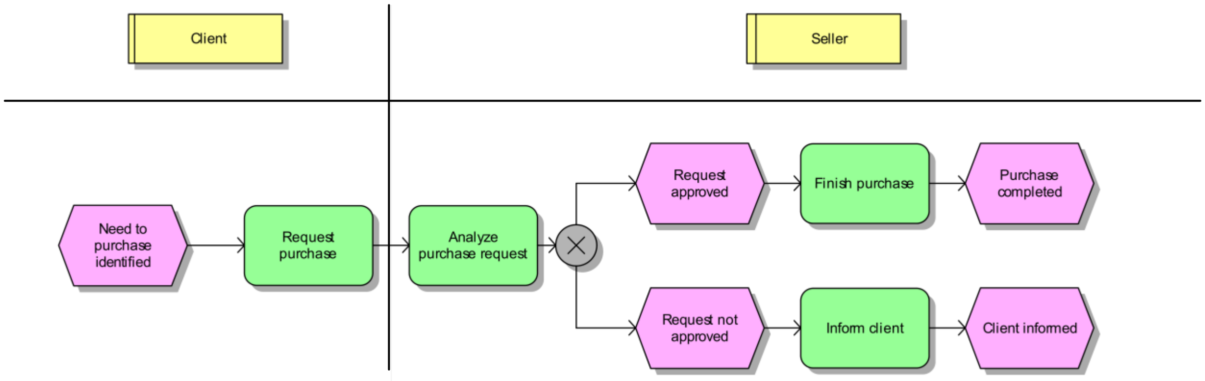

Figure 2 - Example of Business Process modeling in EPC.

a business process or may be created by actors external to the process [6][26]. Figure 2 presents several instances of the *Event* metaclass, which are depicted as hexagons (e.g. “Need to purchase identified”).

The *activates* and *creates* meta-associations between the elements *Function* and *Event* are used to indicate, respectively, that a task is triggered by one or more events and that one or more tasks can trigger one or more events. In Figure 2, instances of these meta-associations are illustrated by arrows connecting tasks and events.

The *Rule* element controls the flow of the process model on the basis of the results and effects of its preceding tasks. In this context, the *is evaluated by* meta-association is used to indicate which events the element *Rule* will select. The *activates* meta-association between the elements *Rule* and *Function* specifies which *Functions* are activated by the *Rule*. The *leads to* meta-association represents the connection between the task which precedes the rule (*Rule*) and the events (*Event*) which are created by the task. The meta-association *links* of the *Rule* element is used to specify behavioral rules of the business process of higher complexity, i.e., rules which can be constructed by the combination of more elementary rules. All these different relationships are represented as arrows in an EPC diagram.

We consider in our analysis three basic types of *Rules*: *XOR*, *AND* and *OR* [6]. These rules (as explained in Table 1) are used for creating “joins” and “splits” in the business process. A “join” unites several incoming branches of the business process into one single outgoing branch. The “split” divides an incoming branch into two or more outgoing branches. Figure 2 presents an example of an XOR “split”, indicating that the “Analyze purchase” can be followed by either “Finalize purchase” or “Inform client”, but not both.

Table 1 – XOR, AND and OR Rules

Rule	Symbol	Join	Split
OR	∨	Any incoming branch, or a combination of incoming branches, will initiate the outgoing branch	One or more outgoing branches will be enabled after the incoming branch finishes
XOR	×	One, and only one, incoming branch will initiate the outgoing branch	One, and only one, outgoing branch will be enabled after the incoming branch finishes
AND	∧	The outgoing branch initiates only after all the incoming branches have been executed	Two or more outgoing branches are enabled after the incoming branch finishes

3. ONTOLOGICAL FOUNDATIONS

To perform ontological interpretation and analysis, we will use concepts of a philosophically and cognitively well-founded reference ontology called Unified Foundational Ontology (UFO) discussed in depth in [12] [13][14]. UFO has been chosen because it unifies several foundational ontologies and has been successfully employed to evaluate, re-design and integrate the models of conceptual modeling languages as well as to provide real-world semantics for their modeling constructs [14]. For example, in [12], a complete evaluation and re-design of the UML 2.0 metamodel using UFO is presented. Figure 3 shows a fragment of this ontology focusing on those concepts which are more relevant for the purposes of this article.

First, UFO distinguishes between conceptual entities called *universals* and *individuals*. The notion of universal underlies the most basic and widespread constructs in conceptual modelling. Universals are predicative terms that can possibly be applied to a multitude of individuals, capturing the general aspects of such individuals. Individuals are entities that exist possessing a unique identity.

Further, UFO is based on the distinction between the concepts of Endurants and Events (also known as Perdurants). Endurants are individuals said to be wholly present whenever they are present, i.e., they are in time, in the sense that if we say that in circumstance *c1* an endurant e has a property *P1* and in circumstance *c2* the property *P2* (possibly incompatible with P1), it is the very same endurant e that we refer to in each of these situations. Examples of endurants are a house, a person, the moon, a hole, an amount of sand. For instance, we can say that an individual John weights 80kg at *c1* but 68kg at *c2*. Nonetheless, we are in these two cases referring to the same individual John.

Events (Perdurants) are individuals composed by temporal parts, they *happen in time* in the sense that they extend in time accumulating temporal parts. An example of an Event is a business process. Whenever an Event occurs, it is not the case that all of its temporal parts also occur. For instance, if we consider a business process “Buy a product” at different time instants when it occurs, at each of these time instants only some of its temporal parts are occurring.

Events can be either atomic (Atomic Event) or complex (Complex Event), depending on their mereological structure, i.e., whilst atomic events have no sub-parts, complex events are aggregations of at least two events (which in turn can be atomic or complex). Events are possible transformations from a portion of reality to another, i.e., they may change the reality by changing the *state of affairs* from one (pre-state) Situation to a (post-state) Situation.

Events are ontologically dependent entities since they existentially depend on their participants in order to exist. Take for instance the event *e*: the stabbing of Caesar by Brutus. In this event we have the Participation of Caesar himself, of Brutus and of the knife. In this case, *e* is composed of the individual participation of each of these entities. Each of these participations is itself an event that can be Complex Event or Atomic Event but which existentially depends on a single Substantial (a substantial is an endurant that does not depend existentially on other endurants).

Situations are special type of Endurants [7][13][14]. These are complex entities that are constituted by possibly many endurants (including other Situations). Situations are taken here to be synonymous to what is named *state of affairs* in the literature, i.e., a portion of reality which can be comprehended as a whole. Examples of situations include *“John being with fever and influenza”, “John being in the same location as Paul while Mary is in the same location as David”, “Mary being married to Paul who works for the University of Twente*”. A Situation Universal [7] characterizes Situations with common properties.

UFO also adds distinctions concerning the intentionality of events to this basic core. Examples include the concepts of Action, Action Universal, Action Contribution and Agent.

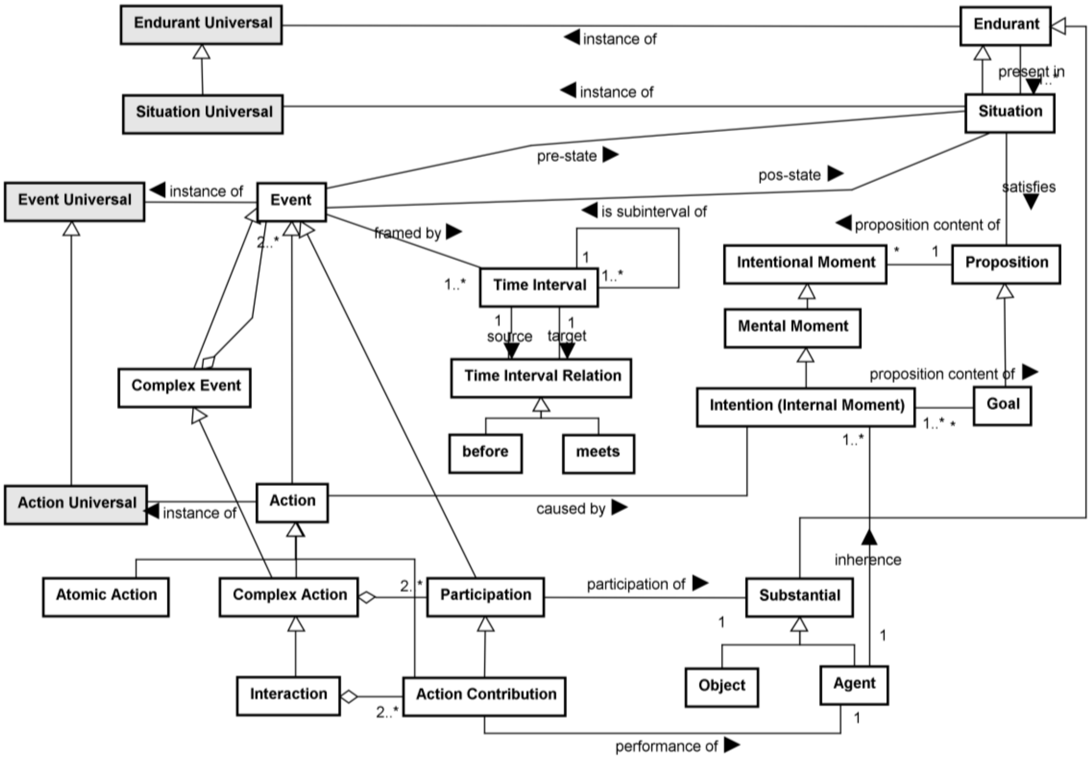

Figure 3 - Fragment of the Unified Foundational Ontology (UFO).

Actions are intentional events, i.e., events which instantiate a Plan (Action Universal) with the specific purpose of satisfying (the propositional content of) some Internal Commitment of an Agent. Only agents (entities capable of bearing intentional moments) can perform Actions. As events, actions can be atomic (Atomic Action) or complex (Complex Action). A Complex Action is composed of two or more Participations.

Participations can themselves be intentional (i.e., Actions) or non-intentional Events. For example, the stabbing of Caesar by Brutus includes the intentional participation of Brutus and the non-intentional participation of the knife. In other words, we take that it is not the case that any participation of an agent is considered an action, but only those intentional participations–called Action Contributions. A Complex Action composed of action contributions of different agents is termed an Interaction. Two artists collaborating to create a sculpture is an example of an interaction, and so is a dialogue between any two agents, e.g., business units of an organization.

UFO defines a conceptual space of time as being a structure "composed of" Time Intervals. A Time Interval in turn is "composed of" Time Points. Time Points could be represented as real numbers and Time Intervals as sets of real numbers. For ordered temporal structures, we have considered here the so-called *Allen Relations* [1] which can hold between intervals, and from which corresponding relations between events can be derived (we consider the "before" and "meets" relations).

4. ONTOLOGICAL ANALYSIS OF THE ARIS EPC METAMODEL

According to [23] and [28], a business process can be defined as a collection of interrelated organizational tasks, initiated in a response to an event, which aim at achieving one or more organizational goals. In other words, a business process describes a type of organizational task that must be performed to achieve one or more organizational goals. Since EPCs are used for business process modeling, we can say that, collectively, the elements of an EPC can be interpreted as a Complex Action Universal of UFO.

4.1 Ontological Analysis of the Function Element

According to [26], a *Function* can be defined in several abstraction and refinement levels. Therefore, we can interpret the *Function* element in different ways: **(i)** Atomic Action Universal, **(ii)** Complex Action Universal or **(iii)** Interaction Universal.

The interpretation of the *Function* element as an Atomic Action Universal occurs when the *Function* element represents an atomic action (without any further level of refinement) and is performed by only one agent (*Participant*).

The interpretation of the *Function* element as a Complex Action Universal occurs when the *Function* element represents a non-atomic action which is composed by two or more Action Universals and is performed by only one agent (*Participant*).

The interpretation of the *Function* element as an Interaction Universal occurs when the *Function* elements represent an Action that is perfomed by two or more agents.

The different ontological interpretations for the *Function* element, suggests an issue of incompleteness in the language, i.e, there are ontological entities in the reference ontology that are not represented in the language. In other words, the *Function* element is semantically overloaded, which may hinder the clarity of the modeling language for certain purposes. For example, the lack of a modeling primitive for interaction often leads to informal labeling strategies for functions (activities) used to model interactions (which start with keywords such as “Inform” and “Notify”.) An example solution for this problem would be the creation of new notational elements to represent alternative ontological interpretations for the *Function* element.

The existence of a *carries out* relationship between a *Participant p* and a *Function f* indicates that an agent will intentionally participate of an organizational task represented by *f*, through an Action Contribution. Depending on the metaclass which specializes *p*, the agent will be an instance of the Universal represented by *p* (in the case of *Position*, *Person Type, Employee Variable and Organizational Unit Type)* or will be the individual agent denoted by *p (*in the case of *Person, Organization Unit* and *Group).*

The meta-association *is predecessor of* defined for the *Function* element represents a temporal relationship among two or more tasks of the business process. In other words, this relationship defines that for one or more tasks to be performed, it is necessary that the antecedent tasks have been concluded. This relationship also describes that a task “produces” a post-state which is a pre-state (Situation) of a successor task (although this state is not explicitly modeled as an *Event*).

The *is precedessor of* meta-association is interpreted as a Time Relation in UFO, more specifically, a Before Relation or a Meets Relation.

4.2 Ontological Analysis of the Event Element

The *Event* element is used in two different ways in EPCs: **(i)** to represent a situation of interest (informally, a “state” of interest) or **(ii)** to represent an arbitrary action that influences the execution of the business process (informally, an “event”).

In the first case (event as a “state”), the *Event* element of the ARIS Method must be interpreted as a Situation Universal. During the execution of the business process, instances of this Universal will be the preconditions (pre-situations) which enable organizational tasks, or postconditions (post-situations) which result from them. Since events in the ARIS Method can involve an arbitrary textual description, there are no rules prescribing which type of entities can be present in (present in) some situation.

A *Start Event* is denoted by the absence of a task which precedes it. The *Start events* represent the situations which enable the business process as a whole. The tasks which enable this situation, in turn, are not part of the business process at hand (which means that the start event it is never the target of a *creates* relationship in the scope of the business process).

An *Intermediate Event* is used between tasks or between tasks and rules. An *Intermediate Event* represents a situation which is obtained as the result of one or more tasks (*Functions*) of the business process. This situation enables one or more tasks (through the *activates* relationship).

An *End event* is represented by the absence of a task which succeeds the business process, i.e., and *End event* is never used as a source for *activates* and *is evaluated by* relationships.

In the second case, the *Event* element of the ARIS Method is used to represent occurrences of events (both intentional and non-intentional) which depend on Endurants. For example, “security alarm”, “client request” or “change of room temperature”. In doing so, the *Event* element of the ARIS Method is interpreted as an Event Universal in UFO. Since events in the ARIS Method can involve arbitrary textual descriptions, there are no rules prescribing which type of entities can participate in an *Event* (i.e., what type of Participations are allowed). Therefore, the interpretation of the *Event* modeling element cannot be more specific than Event Universal.

Under the first interpretation for *Event* (when it is interpreted as a Situation Universal), the *activates* meta-association represents that one or more events are pre-states (pre-state Situation) necessary for one or more tasks. In other words, to initiate a task, it is necessary that the situation holds. The *creates* relationship between *Function* and *Event* represent that one or more tasks brings about situations (post-state of Function). These situations can then be pre-state for other business process tasks.

If the *Event* element of the ARIS Method is interpreted instead as an Event Universal, the *activates* meta-association then represents a temporal relation, more specifically, a Before or Meets relation between the Events represented by *Event* and *Function*. In other words, this meta-association defines that for one or more tasks to be performed, it is necessary that the predecessor events to these tasks have already occurred. The *creates* relationship represents then the inverse of the *activates* relationship (also expressing a temporal relation among Events).

The existence of these alternative ontological interpretations for the *Event* element of the ARIS Method constitutes a problem of semantic overload. A possible solution to this problem is the creation of new notational elements that represent the alternative ontological interpretations for the *Event* element. For example, one could create elements in the language to distinguish intentional and non-intentional events. Another possible recommendation is the exclusive use of *Events* to represent “states” (as suggested by [6]). By following one of these recommendations, this specific case of semantics overload would be eliminated.

Due to space restrictions, we assume the first interpretation of *Event* (*Event* as a Situation Universal) in Section 4.3, which discusses the ontological analysis of the *Rule* element.

4.3 Ontological Analysis of Rule Element

The set of *Rules* of a business process specifies which tasks must be carried out based on a given *state of affairs*. The *state of affairs* for the business process is formed by the internal events (*Events* produced by the tasks) and the events external to business process (*Events* produced by other processes or entities).

The *Rule* element allows one to compose complex events from the union of internal and/or external events. These complex events can be a requirement for a particular task to occur. For example, the notational element *AND* is used to compose a complex event, from two or more events of some business process. This complex event created by the notational element *AND* can be a pre-state for one or more tasks of the business process. The self-relationship *links* of the *Rule* element can also be used to create complex events through the combination of *Rules*.

Differently from the other elements of the business process metamodel of the ARIS Method, no direct interpretation to the *Rule* element can be offered in terms of the ontological categories of UFO. This occurs because *Rules* describe relations between Situation Universals and the current perdurant fragment of UFO focuses on the relations between individuals. However, since these relations between Universals are used to determine the possible establishment of relations between individuals, we can offer an ontological account for *Rules*. If we assume that the *Event* element represents a Situation Universal, then we have the following interpretations (considering a pattern formed with two *Events*, X and Y which are *evaluated by* a rule which *leads* to the *Event* Z) for the tree basic *Rules* (AND, OR, XOR): **(i) X AND $Y \rightarrow Z$**: this means that every instance of Z has as pre-situation one instance w of a Situation Universal W and every instance of w is the mereological sum of an instance of X and one instance of Y (both situations). **(ii) X OR $Y \rightarrow Z$**: this means that every instance of Z has as pre-situation one instance w so that w is an instance of X, or w is an instance of Y, or there are situations w' and w'' such that w' is an instance of X, w'' is an instance of Y, and w is a mereological sum of w' and w''; **(iii) X XOR Y → Z**: this means that every instance of Z has as pre-situation one individual w such that w is either an instance of X or Y but not both, since X and Y are disjoint Situation Universals.

5. DISCUSSION AND RELATED WORK

The study which is most closely related to our work was conducted by Green and Rosemann and presented in [5][10][11].Green and Rosemann discuss an ontological analysis of EPC models based on the BWW ontology [36]. The differences between the two approaches are significant because of the choices in the different foundational ontologies employed and the mapping choices employed in the analysis; however, we cannot provide a full-length comparison here due to space constraints. Just to illustrate some of these differences, in our approach, events (and consequently functions) are treated as first-class citizens and, as a consequence, the relations involving functions (decomposition, object participation) can be explicitly defined. In contrast, in Green and Rosemann's work, by interpreting the Function element as a formal mapping between two values in a conceptual space of states, the authors do not provide an ontological explanation for: (i) the participation of agents in processes; (ii) the notion of time associated with the execution of an organizational task and (iii) the definition of complex functions through mereological relations between functions and sub-functions. In another aspect, UFO, but not the BWW ontology, makes an explicit distinction between unintentional events and (intentional) actions. To understand business processes and notions such as services as social phenomena, the notions of goals and commitment are of fundamental importance [28]. This requirement places an approach founded on an ontology in which social reality is treated in an explicit manner in clear advantage.

Other authors, using the term "Semantic Business Process Management" [31], propose to annotate the elements of a business process models with additional information ("semantic annotation"), often by referring to entities of a business domain ontology to define the goals, preconditions and effects of activities in a business process model. We regard our approach as a pre-requisite for sound "semantic annotation", when both the elements of the annotated model and the elements of the annotation are interpreted in terms of a reference ontology and given a harmonious semantic foundation.

The approach sketched initially in [31] was further developed in [32] where the authors proposed a semantic annotation approach to EPCs based on an ontology that mirrors a small fragment of the EPC metamodel (thus not clarifying the semantics of the EPC modeling elements).

Further, the SUPER project has investigated a notion of Semantic EPCs or sEPC ([8], [9]). Their approach is based on a transformation of Semantic EPCs into the Business Process Modeling Ontology (BPMO[30][34]), which they propose as an abstraction of process modeling techniques (including EPCs and OMG's BPMN). BPMO, in its turn, specializes concepts of an upper-level ontology called UPO which is inspired in the well-founded DOLCE [3] ontology. The relation between Semantic EPCs and its semantic foundation is thus rather indirect, and, as a consequence their work cannot elucidate the nuances in the interpretation of EPCs.

In [2], we have provided a semantic foundation for role-related concepts in several Enterprise Modeling approaches, including the ARIS framework. In that work, we have provided an ontological interpretation for the concepts of *Organizational Unit*, *Organizational Unit Type*, *Position*, *Employee* (same as *Person*) and *Role* (same as *Person Type*), also by employing the UFO foundational ontology. These concepts are subsume in this paper by the *Participant* element. Through ontological analysis it was possible to propose a revision of the ARIS *Position* concept unifying it with the notion of role. This leads to a more parsimonious set of role-related concepts in ARIS and, ultimately, more parsimonious models.

6. CONCLUSIONS AND FUTURE WORK

The ontological analysis presented in this paper provides a better understanding of EPCs with the support of a foundational ontology. An immediate benefit of our ontological analysis is related with the development of business process models with well-defined real-world semantics. Since EPCs are used to capture and communicate organizational knowledge (and possibly shape process-oriented information systems) these models must be free from ambiguities. Ontological interpretation adds to existing formal execution semantics definition, leading to a comprehensive semantic foundation which clarifies the relation between models and the corresponding modeled phenomena (business activities and processes), while also enabling simulation, automated analysis and execution.

The ontological analysis has allowed us to reveal problems of usage of certain modeling elements in EPC models (e.g., in the case of the *Event* element). This analysis has also addressed a problem informally identified in the literature ([6], p. 111) in a systematic manner.

Our next steps will focus on an ontological analysis of the ARIS notations used for modeling the organizational domain (the Organization Chart), and the one used for capturing the detailing of activities (the Function Allocation Diagram - FAD).

ACKNOWLEDGEMENTS

This work has been supported by CNPq and by FAPES in the scope of the INFRA-MODELA project.

7. REFERENCES

[1] Allen, J.F. 1983. Maintaining Knowledge About Temporal Intervals. In Communications of the ACM, Vol.26, no. 11.

[2] Almeida, João Paulo A., Guizzardi, G., and Santos Jr., P.S. 2009. Applying and extending a semantic foundation for role-related concepts in enterprise modelling. In Enterprise Information Systems, 253-277.

[3] Bottazi, E. and Ferrario, R. 2006. Preliminaries to a DOLCE Ontology of Organizations. In Journal Business Process Integration and Management. Vol.1. 64–74. 2006.

[4] Davidson, D. 2001. Essays on Actions and Events. In Oxford: Oxford University Press, Ed. 2nd.

[5] Davis, I.G., Rosemann, M., and Green, P. F. 2004. Exploring proposed ontological issues of ARIS with different categories of modellers. In: Proc. of the Australasian Conf. on Information Systems (Hobart, Australia, 2004).

[6] Davis, R. 2001. Business Process Modeling with ARIS - A Pratical Guide. Springer.

[7] Dockhorn, P. 2006. Situations in Conceptual Modeling of Context. In Proc. 2nd Int'l W. on Vocabularies, Ontologies and Rules for The Enterprise (VORTE'06), IEEE CS Press.

[8] Filipowska A., Kaczmarek, M., Markovic, I., Kowalkiewicz, M., and Zhou, X. 2009. Organizational Ontologies to Support Semantic Business Process Management. In SBPM.

[9] Filipowska, A., et al., 2009: Organisational Ontology Framework for Semantic Business Process Management. In Proc. 12th Int'l Business Information Systems Conference (BIS 2009). Springer LNBIP, Vol. 21, 1-12.

[10] Green, P. F. and Rosemann, M. 2005. Ontological Analysis of Business Systems Analysis Techniques: Experiences and Proposals for and Enhanced Methodology. In Business Systems Analysis with Ontologies.

[11] Green, P., Indulska, M., and Rosemann, M. 2004. A Reference Methodology for Conducting Ontological Analyses. In Proceedings of the 23rd International Conference on Conceptual Modelling (ER 2004), 8-12.

[12] Guizzardi, G. 2005. Ontological Foundations for Structural Conceptual Models. Ph.D. Thesis. CTIT PhD-thesis No. 05-74., University of Twente. The Netherlands. TI-FRS No. 15.

[13] Guizzardi, G. and Wagner, G. 2004. On A Unified Foundational Ontology and some Applications of it in Business Modeling. In 16th Int'l Conf. on Advances in Information Systems Engineering (Latvia). CAiSE 2004. Open INTEROP Workshop on Enterprise Modelling and Ontologies for Interoperability.

[14] Guizzardi, G., Falbo, R. A., and Guizzardi, R. S. S. 2008.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 (IDEAS'2008) (Recife, Brazil, 2008).

[15] Guizzardi, G., Ferreira Pires, L., and Van Sinderen, M. 2005. An Ontology-Based Approach for Evaluating the Domain Appropriateness and Comprehensibility Appropriateness of Modeling Languages. In ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems. Lecture Notes in Computer Science LNCS 3713. Springer-Verlag. Montego Bay. Jamaica. 2005

[16] Heller, B. and Herre, H. 2004. Ontological categories in gol. Axiomathes – Kluwer Publishers, v. 14, 71–90.

[17] Kern, H. and Kühne, S. 2007. Model Interchange between ARIS and Eclipse EMF. In The 7th OOPSLA Workshop Domain-Specific Modeling, Computer Science and Information System Reports. Technical Reports, University of Jyväskylä at Finland.

[18] Kindler, E. 2006. On the Semantics of EPCs: A Framework for Resolving the Vicious Circle. Technical Report. University of Paderborn, Germany.

[19] Mendling, J. 2009. Metrics for Process Models, Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. 17-57.

[20] Mendling, J. and Nüttgen, M. 2004. Transformation of ARIS Markup Language to EPML. In Proc. of the 3rd GI Workshop on Event-Driven Process Chains.

[21] Mendling, J., Neumann, G., and Nüttgens M. 2005. Yet Another Event-Driven Process Chain (Extended Version). Technical Report. University of Economics and Business Administration at Vienna, JM-2005-05-27.

[22] Mylopoulos, J., and Castro, J. 2000. Tropos: A Framework for Requirements-Driven Software Development. In Information Systems Engineering: State of the Art and Research Themes. Springer-Verlag. June. 2000, 261-273.

[23] Ouyang, C., Dumas, M., Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., and Mendling, J. 2008. From Business Process Models to Process-oriented Software Systems: The BPMN to BPEL Way. In ACM Transactions on Software Engineering and Methodology.

[24] Rosemann, M., and Van der Aalst, W.M.P. 2007. A Configurable Reference Modelling Language. Inf. Syst. 32. 1-23.

[25] Santos Jr., P. S., Almeida, J. P. A., and Pianissolla, T. L.. Uncovering the Organizational Modelling and Business Process Modelling Languages in the ARIS Method, International Journal of Business Process Integration and Management (IJBPIM), to appear.

[26] Scheer, A.W. 1999. ARIS – Business Process Modeling. Third Edition. Springer.

[27] Shanks, G., Tansley, E., and Weber, R. 2003. Using Ontology to validate conceptual models. In Communications of the ACM 46. p. 85-89

[28] Sharp, A. and McDermott, P. 2000. Workflow modeling: Tools for Process Improvement and Application Development.

[29] Super. 2008. sBPMN and sEPC to BPMO Translation. Version 1.0. April, 2008. http://www.ip-super.org/res/Deliverables/M24/D4.5.pdf.

[30] Super. 2009. Process Ontology Stack. Evolved Version. Version 1.0. March, 2009. http://www.ip-super.org/res/Deliverables/M36/D1.5.pdf

[31] Thomas, O. and Fellmann, M. 2006. Semantic Event-Driven Process Chains. In Workshop SBPM (Budva, Montenegro, 2006). ESWC 2006.

[32] Thomas, O. and Fellmann, M. 2007. Semantic EPC: Enhancing Process Modeling Using Ontology Languages. In Proc. Workshop on Semantic Business Process and Product Lifecycle Management (SBPM) 2007.

[33] Van der Aalst, W.M.P. 1998. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers.

[34] Van der Aalst., W.M.P. 1998. Formalization and Verification of Event-driven Process Chains. Computing Science Reports. Eindhoven University of Technology at Eindhoven.

[35] Van Dongen, B.F., Van der Aalst, W.M.P., and Verbeek., H.M.W. 2005. Verification of EPCs: Using Reduction Rules and Petri Nets. In Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE 2005) LNCS. Vol. 3520, 372-386, Springer.

[36] Wand, Y. and Weber, R. 1989. An ontological evaluation of systems analysis and design methods. In Information System Concepts: An In-Depth Analysis. Elsevier Science Publishers B.V., North-Holland.