Journal of Science Research (2013) Vol. 12: 275-283

A Dynamic Ontology for Shared Representation of Knowledge of Establishments

Ayorinde, I.T.* and Akinkunmi, B.O.

Abstract

Explicit specifications of domain conceptualizations called ontologies, are essential for the development and use of intelligent systems. Implicit and shared representations of knowledge help to adequately and consistently use the required knowledge to solve problems in an establishment. This paper builds a dynamic ontology for shared representation of an establishment and discusses the major concepts and structures that must be identified and represented in such an establishment while building the dynamic ontology. It has approached the problem of capturing the dynamic essence of an establishment by addressing the representational problems involving occurrents (states, events and processes). We recommend that states be represented using the method of temporal arguments, while events should be represented using the Davidson style reification. The language of representation is first order logic. The temporal nature of events and processes is elucidated and the causation relation between processes and events were formalized. Events in this domain are instantaneous while processes take place over intervals. Other structures that make up our ontology include temporal definitions and temporal classifications. The ontology enables us not just to answer queries with what is explicitly represented in it but also with what is implied by the shared representation of the establishment.

Key words: Ontology, States, Events, Processes, Shared Representation.

Introduction

Ontology is the explicit formal specifications of the terms in a domain and the relationship between them. An ontology actually defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and the relationship between them [1, 2]. Figure 1 gives a clearer view of the meaning of an ontology [3].

Ontologies are useful in many ways for human understanding and interaction. For example, they can serve as the embodiment of (and reference for) a consensus reached by a professional community (e.g., physicians) on the meaning of a technical vocabulary that is to be used in their interactions. They can also serve as sets of codes and standard vocabulary with which queries and assertions are exchanged among agents [4]. Such codes and standards can be used for carrying out shared representation in establishments such as the civil service, institutions, private and public organizations, among others.

Ontologies are particularly useful in knowledge sharing and reuse. If two agents make the same ontological commitment, then they can exchange knowledge.

Since implicit and shared representation of organizational knowledge are hard to come by in most establishments which make it impossible for them to adequately and consistently use the required knowledge to solve problems in the establishment, there is the need to capture a shared representation of such establishments and this can be done with formalized ontologies.

Ontology produces theories about the world, formalized in some logical language. The virtue of formalization is first of all that of enforcing a certain degree of clarity. Another virtue is that it makes theories readily accessible, evaluable, and re-usable by other community of researchers. Additionally, formalization makes it possible for us to exploit some of the power of logic when using ontologies in reasoning systems [5].

Ayorinde, I.T.* and Akinkunmi, B.O.
**Department of Computer Science,*
University of Ibadan, Ibadan, Oyo State,
Nigeria. +2348035289814
Alternative Email: lolaayorinde@gmail.com

ISSN 11179333

Fig. 1: Diagrammatic representation of ontology.
Source: Minz 2006

This paper builds a dynamic ontology for shared representation of establishments and has approached the problem of capturing its dynamic essence by addressing the representational problems involving occurrents (states, events and processes). Our objective is to elucidate the temporal nature of events and processes in this domain and formalize the causation relation between them.

In order to introduce precision to our shared representation, we need a precise formalism that enables us to reason with our representation. As such, our ontology will consist of a logical formalization of a coherent domain. Static ontologies on the other hand capture the state of affairs in an organization without taking cognizance of the dynamics of the organization.

Review of Related Concepts

Overview of Ontology

Ontology is an interdisciplinary field involving both philosophy and science [6]. Ontology is not a discipline which exists separately and independently from all the other scientific disciplines and also from other branches of philosophy. Rather, ontology derives the general structure of the world; it obtains the structure of the world as it really is from knowledge embodied in other disciplines [7]. Ontology mirrors, so to speak, the level of our knowledge of the world at any given time.

Ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. It is used to reason about the entities within that domain, and may be used to describe the domain. An ontology can also be defined as a "formal, explicit specification of a shared conceptualization". An ontology provides a shared vocabulary, which can be used to model a domain; that is, the type of objects and/or concepts that exist, and their properties and relations. Figure 2 shows different kinds of ontology that can be developed according to their level of dependence on a particular task or point of view [8]. Top-level ontologies describe very general concepts like space, time, matter, object, event, action, etc., which are independent of a particular problem or domain: it seems therefore reasonable, at least in theory, to have unified top-level ontologies for large communities of users. Figure 3 shows a simple top-level ontology [9].

Continuants are also known as Endurants while Occurrents are known as Pedurants. Occurrents are all bound in time. This means that each portion of the time during which an occurrent occurs can be associated with a corresponding temporal portion of the occurrent. This is because occurrents exist only in their successive temporal parts or phases. Some occurrents – for example the initial and terminal boundaries of processes, are instantaneous and are hence referred to as events. The term perdurant is more precisely used for those occurrents which persist (perdure) in time, in other words, for those which are extended and not instantaneous. The term 'process' is used for temporally extended occurrents. Entities which exhaust themselves in single instants of time are also referred to as 'events'.

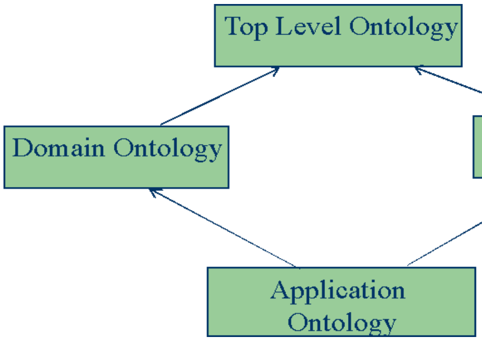

Fig. 2: Kinds of ontologies (Arrows represent specialization relationships).
Source: Guarino 1998

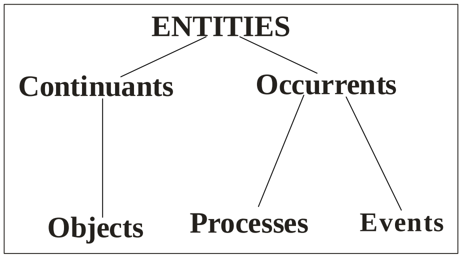

Fig. 3: Top-level ontology.
Source: Galton 2006

Domain ontologies and task ontologies describe, respectively, the vocabulary related to a generic domain (like medicine, or automobiles) or a generic task or activity (like diagnosing or selling), by specializing the terms introduced in the top-level ontology.

Application ontologies describe concepts depending both on a particular domain and task, which are often specializations of both the related ontologies. These concepts often correspond to roles played by domain entities while performing a certain activity, like replaceable unit or spare component.

Dynamic and Static Ontologies

The concepts used for knowledge representation classify ontology into different ontological categories. Static ontologies describe static aspects of the world, i.e., what things exist, their attributes and relationships while a dynamic ontology, on the other hand, describes the changing aspects of the world in terms of states, state transitions and processes [10].

Most knowledge representation frameworks assume that the world is populated by entities that are endowed with a unique and immutable identity, a lifetime, a set of attributes, and relationships to other entities. Hayes [11] offers an ontology for different classes of applications modelling of material substances where entities (say, a liter of water and a pound of sugar) can be merged resulting in a different entity. Static ontologies are not trivial. For certain applications, it is useful to distinguish between different modes of existence for entities, including physical existence [8].

Various flavors of finite-state machines and Petri nets have been offered since the 1960s as appropriate modelling tools for dynamic discrete processes. Such models are well understood and have been used extensively to describe real-time applications in telecommunications and other fields. Statecharts constitute a more recent proposal for specifying large, finite-state machines [12]. A statechart is also defined in terms of states and transitions, but more than one state may be on at any one time, and states can be defined as AND or OR compositions of other statecharts. As a result, statecharts have been proven much more effective in defining and simulating large, finite-state machines compared with conventional modelling methods.

The introduction of temporal reasoning in the artificial intelligence literature [13] has made it possible to use time explicitly to model the dynamic aspect of a system. In Basic Formal Ontology (BFO), a series of sub-ontologies (most properly conceived as a series of perspectives on reality) are SNAP and SPAN [14]. These two types of ontology (i.e. SNAP and SPAN), as advocated by Barry Smith and collaborators and according to Grenon and Smith [15], “A good ontology must be capable of accounting for spatial reality both synchronically (as it exists at a time) (SNAP) and diachronically (as it unfolds through time) (SPAN)”.

SPAN is the ontology of those entities which unfold themselves through time in their successive temporal parts. (SNAP entities do not have temporal parts). Occurrents are extended in time; they have temporal parts, and thus they can be partitioned via partitions of the temporal dimension. Time itself is a constituent of the reality that is captured by SPAN and this makes SPAN very suitable for developing this ontology since “dates of occurrences or events” are very paramount here.

SNAP is a series of snapshot ontologies indexed by times while SPAN - a single videoscopic ontology. Each Snap is an inventory of all entities existing at a time. Span is an inventory (processory) of all processes unfolding through time. Each snapshot ontology represents an assay of the entities existing at some given present instant. *Span* is a partition of the totality of processes. Processes are invisible in the snapshot view; substances are invisible in the span view. Both *SNAP* and *SPAN* serve as basis for a series of sub-ontologies at different levels of granularity. The same portion of reality may appear at a plurality of levels of granularity. Thus, masses at one level may be aggregates at another level. What counts as a unitary process at one level may be part of a process-continuum at another level. Static ontology imbibes Snap (or Snapshot Ontology) which makes use of continuants or endurants while dynamic ontology imbibes Span ontology which makes use of occurrents or Pedurants.

Language and Notation

The representational language used to build this ontology was many sorted first order logic (MSFOL). This language was chosen because of its expressiveness and efficiency. The sorts are the domain entities, time points, time intervals and events. First-order logic is a formal logical system used in mathematics, philosophy, linguistics and computer science. It goes by many names, including: first-order predicate calculus, the lower predicate calculus, quantification theory, and predicate logic.

First-order logic is distinguished from propositional logic by its use of quantifiers ($\forall$ and $\exists$). Each interpretation of first-order logic includes a domain of discourse over which the quantifiers range. It allows reasoning about properties that are shared by many objects, through the use of variables and more flexible and compact representation of knowledge.

First-order logic also satisfies several metalogical theorems that make it amenable to analysis in proof theory. It is widely used in the theory of knowledge representation. MSFOL-based language is a pair (A, W), where A is an alphabet of symbols and W is a set of syntactically well-formed formulae of the symbols of A which comprises of the following:

- Logical symbols: punctuation symbols such as parentheses: (,) and sentential connective symbols: **¬, ∨, ∧, →, ↔.**
- Sorts: a non-empty finite set of symbols S, each of which represents a simple sort and to which the following rules are applied when the sorts are defined: (a) a simple sort is a sort, (b) if S_1 and S_2 are sorts, then $S_1 \wedge S_2$, $S_1 \vee S_2$ and $\neg S_1$ are also sorts.
- Quantifier symbols: for each sort S, there is a universal quantifier symbol **∀s** and an existential quantifier symbol **∃s**.
- Constant symbols: for each sort S, there is a finite set of constant symbols, each of which is said to be of sort S [16].

Temporal Ontology

An approach to engineering ontologies, determined from the literature [17] was followed in building this dynamic ontology. This begins with defining the ontology's requirements; this is in the form of questions that the ontology must be able to answer. We call this the competency of the ontology. Secondly, the terminology of the ontology is defined to include its objects, attributes, and relations from the domain. The third step was the specification of the definitions and constraints on the terminology, where possible. The fourth step is the implementation. Implementation here is done using the Prolog programming language.

Our domain is the establishment system of the Oyo State Civil Service. In order to capture the dynamics of our domain, we focus on identifying the continuants in the domain and their temporal properties i.e. states, events and processes. In addition to this, we make use of domain constraints that have temporal components. To capture the dynamic structure of the ontology, a temporal framework based on Allen's interval logic and McDermott's logic of time points is used [18, 19]. An interval is taken to be a pair of time points. There are different ways of describing and classifying what goes on in time. It is common to distinguish three main categories: states, processes and events. Each of these characterizes a situation from a different point of view: A state "abstracts away from any changes that are taking place and focuses on the unchanging aspects of a situation". A process "focuses on ongoing change as it proceeds from moment to moment, not as a completed whole" while an event is "an episode of change with a beginning and an end, considered as a completed whole" [1]. Since this research work deals with capturing the dynamics of an organization, continuants such as events and processes along with states must be adequately captured and represented.

We deal with time points as primitive elements of the temporal ontology. Intervals are regarded as pairs of time points. In the domain, a time point is reckoned as calendar dates. For example: *an appointment takes effect from 12th January 200,3* etc. Time intervals are reckoned as pairs of time points e.g. (t_1, t_2) denote an interval starting at t_1 and ending at t_2, where t_1 and t_2 are time points. The overall structure of time is assumed to be linear (as against branching time). The predicates of the language denote the relations in the domain. The definitions give a description of the concepts while axioms describe the domain constraints.

Our basic Ontology of occurrents consists of states, events and processes. States hardly need to be instantiated often but events need to be instantiated because two instances of the same event may have different qualifications. Hence, Davidson's instantiation approach is more appropriate for events than for states. For example, two instances of the event "John killed a snake" may differ on the weapon used in both instances. Besides, some state instances may be too long to be of any practical importance e.g. "The car is red". The car may be red for ever. As it turns out in this domain, events are caused by processes. The processes take place over an interval while the events are reckoned as happening instantaneously.

The following sections discuss the examples of states, events and processes that arise in our domain of study.

Formalization of the Conceptualization (Axioms)

States

We represent states using the Method of Temporal Arguments (MTA) [20] e.g. On (a, b, t), On (a, b, (t_1,t_2)). The essence of the method of temporal arguments is its use in every predicate to establish temporal references. It admits, as predicates, ordinary properties and relations. The basic temporal entities referred to by temporal arguments may be either points or intervals. The MTA approach can be very expressive when the more expressive features are chosen and supported by a full set of temporal predicates [21].

Example:

The commissioner in a ministry during a time interval is the political head of the ministry i.e.

$$\forall p, m, t_1, t_2.\ \text{Political-Head}(p, m, (t_1, t_2)) \leftrightarrow \exists m.\ \text{Ministry}(m) \wedge \text{Commissioner}(p, m, (t_1, t_2))$$

The above axiom simply says that for all p, m, t_1 and t_2 where, p is a political head, m is a ministry, $t_1$and t_2 are pairs of time points (or time interval), p is the political head of that ministry if and only if there exists m where m is a ministry and p is the commissioner of that ministry within the same time interval.

We also have definitions, axioms and classifications as instants of state.

Definition

The head of a unit during a time interval is an assistant director.

$$\forall h, u, t_1, t_2.\ \text{Head}(h, u,(t_1,t_2)) \leftrightarrow \exists u.\ \text{Unit}(u) \wedge \text{Member}(h, u, (t_1, t_2)) \wedge \text{Status}(h, \text{Assistant\_Director})$$

The definition given above simply says that for all h and u at time interval t_1,t_2, h is the head of a unit at time interval t_1,t_2, if and only if there exists a unit, u where h is a member of the unit at the same interval of time and the status of h is assistant director.

Axiom

For example, a political office holder cannot be a career civil servant at the same time as shown below:

$$\forall x, t_1, t_2.\ \text{Political\_Office \_Holder}(x, (t_1, t_2)) \rightarrow \neg\, \text{Career\_Civil\_Servant}(x, (t_1, t_2))$$

The above axiom simply says that for all x, (t_1, t_2), where, x is a political office holder within (t_1, t_2), s/he cannot be a career civil servant at the same specified period of time.

Classification Axiom

Civil servants can be categorized into three cadres: Junior Cadre, Executive Cadre and Officer Cadre.

$$\forall x, t_1, t_2.\ \text{Civil\_Servant}(x, (t_1, t_2)) \leftrightarrow \text{Junior\_Staff}(x, (t_1, t_2)) \vee \text{Executive\_Cadre}(x, (t_1, t_2)) \vee \text{Officer\_Cadre}(x, (t_1, t_2)).$$

From the above axiom, a civil servant cannot be in two or more cadres at the same time. S/he can only occupy one of the cadres at a time.

Events

Events are represented using Davidson's style reification [22]. The structure of events is more complex than that of states, requiring the ability to add more attributes.

Example:

The event of an individual converting from one cadre to another occurs at time t i.e.

$\exists e.\ \text{convert}(x, \text{cadre1}, \text{cadre2}, e) \wedge \text{occurs}(e,t)$

Also,
An individual can be in the officer cadre either by direct appointment or by conversion with an evidence that s/he has not left the service. The conversion process here is an event. The axiom for this is shown below:

$\forall x.\ \text{Officer\_Cadre}(x, (t_1, t_2)) \leftrightarrow \exists t.$
$(\text{Appointed\_To\_Officer}(x, t) \vee$
$\text{Converted\_To\_Officer}(x, t) \wedge t \leq t_1\ ) \wedge$
$(\forall t'.\ t \leq t' \leq t_2 \rightarrow \neg \text{Left\_Service}(x, t') \wedge$
$\neg \exists e: \text{Conversion}(x, e) \wedge \text{Time}(e, t')).$

To simplify the above rule, an appointment to the officer cadre can be made directly if an individual's minimum qualification as at the date of first appointment is Bachelor degree. The axiom is shown below:

$\forall x,\ t.\text{Appointed\_To\_Officer}\ (x, t) \leftrightarrow \exists q.$
$(\text{Qualification}(x, q) \wedge$

$\text{Level}(q, \text{First\_Degree}) \wedge \exists t': \text{Date}(q, t') \wedge$
$t' \leq t) \wedge$
$\exists e.\ \text{Appointment}(x, e) \wedge \text{Date}(e,t)\ \wedge \text{Post}(e,$
$\text{officer II}) \vee$

$\text{Post}(e, \text{Officer I}) \vee \text{Post}(e, \text{Senior\_Officer})$
$\vee \text{Post}(e, \text{Principal\_Officer}) \vee$

$\text{Post}(e, \text{Chief\_Officer}) \vee \text{Post}(e, \text{Chief\_}$
$\text{Officer}) \vee$

$\text{Post}(e, \text{Assistant\_Director}) \vee \text{Post}(e,$
$\text{Deputy\_Director}) \vee \text{Post}(e, \text{Director}) \vee$

$\text{Post}(e, \text{Perm\_Sec}))$

Also, an individual in either the junior cadre or executive cadre can convert to an officer cadre if his/her minimum qualification as at the time of conversion is Bachelor degree as shown in the axiom below:

$\text{Convert\_To\_Officer}(x, t) \leftrightarrow (\exists t':$
$\text{Executive\_Cadre}(x, (t', t)) \vee$

$\text{Junior\_Cadre}(x, (t', t))) \wedge$

$\exists q: \text{Qualification}(x,q) \wedge \text{Level}(q,$
$\text{First\_Degree}) \wedge$

$\exists t_1: \text{Date}(q, t_1) \wedge t_1 \leq t \wedge \exists e:$
$\text{Conversion}(x, e) \wedge \text{Time}(e, t) \wedge$

$(\text{Post}(e, \text{officer II}) \vee \text{Post}(e, \text{Officer I}) \vee$
$\text{Post}(e, \text{Senior\_Officer}) \vee$

$\text{Post}(e, \text{Principal\_Officer}) \vee \text{Post}(e,$
$\text{Chief\_Officer}) \vee \text{Post}(e,$
$\text{Chief\_Officer}) \vee$

$\text{Post}(e, \text{Assistant\_Director}) \vee \text{Post}(e,$
$\text{Deputy\_Director}) \vee \text{Post}(e, \text{Director}) \vee$

$\text{Post}(e, \text{Perm\_Sec})).$

Processes

In Ontological literature, there has been two kinds of processes [23].

- Open–ended homogenous activity such as a "pushing a cart" or "writing"
- Routine based processes; the kind that is modelled in *process algebras* and the Process Specification Language, PSL.

In our domain the processes that exist are the routine based processes that may cause certain events to happen. Example is the conversion process that has the following routines:

- The Conversion process begins with an application by the candidate moving from one staff cadre to another (usually based on a newly acquired qualification).
- This is followed by the verification of the new qualification and present status.
- Approval at the necessary boards and committees.
- Final approval by the Head of service.
- If the process succeeds, then a conversion event is deemed to have taken place as a result.

The following axiom simply says that employee x application undergoes a conversion process from cadre1 to cadre2 within a time interval, t_1 and t_2.

$\exists p.\ \text{Conversion\_process}(x, \text{cadre1},$
$\text{cadre2}, p) \wedge \text{Occurs}(p, (t_1, t_2))$

The major difference between processes and events in our domain is that events are instantaneous while processes take place over an interval. Relating processes to event (Causation), we have the following causation axiom:

Conversion process that starts at time t_1 ends with an event in time t_2.

$$\forall x, p.\ \text{Conversion\_process}(x, \text{cadre1}, \text{cadre2}, p) \wedge \text{Occurs}(p, (t_1, t_2)) \wedge \text{Succeeds}(p) \Rightarrow$$

$$\exists e.\ \text{Convert}(x, \text{cadre1}, \text{cadre2}, e) \wedge \text{Occurs}(e, t_2) \wedge \text{Cause}(p, e)$$

The causation axiom given above simply tells us that the conversion process that starts at time t_1 ends with an event in time t_2.

Domain Constraints

There are domain constraints that must be modelled as part of the domain modelling process. These constraints must be dynamic.

Example:

There can only be one person of the status of an assistant director in a unit during a period of time i.e. For all employee p and unit u at time interval t_1, t_2, where p is a member of the unit and the assistant director at the same time, implies that for all employee p_1 who is a member of the same unit at another time interval t_3, t_4, where the two time intervals fall within each other, then, p and p_1 are referring to the same employee. This is shown below:

$$\forall p, u, t_1, t_2.\ \text{Unit}(u) \wedge \text{Member}(p, u, (t_1,t_2)) \wedge \text{Status}(p, \text{Assistant\_director}, (t_1, t_2)) \rightarrow$$

$$\forall p_1.\ \text{Status}(p_1, \text{Assistant\_director}, (t_3, t_4)) \wedge \text{Member}(p_1, u, (t_3, t_4)) \wedge \text{Within}((t_3,t_4), (t_1,t_2)) \rightarrow p = p_1$$

Also, a career civil servant cannot have more than one designation at a time.

$$\forall x, t_1, t_2\text{: Officer\_Cadre } (x, (t_1, t_2)) \rightarrow \neg \exists\ t_3, t_4.\ ((\text{Executive\_Cadre}(x, (t_3, t_4)) \vee$$

$$\text{Junior\_Staff}(x, (t_3, t_4)\ ) \wedge\neg\ \text{Disjoint}(t_1, t_2), (t_3, t_4)))$$

The formalization of this dynamic establishment ontology has made it possible for us to exploit some of the power of logic in reasoning. Also, the prolog implementation of the first order logic axioms enabled us to answer the competency questions for the ontology.

Future Work

This domain gives a variant of the relation between processes and events. This relation contrast with that between a "pushing process" of say a cart p and its moving event e, of the cart [24]. In that case:

$$\text{Cause}(p, e) \wedge \text{Time}(p) = \text{Time}(e)$$

In this example, processes and events happen over intervals. This is unlike our domain where processes take place over intervals and events are instantaneous. Hence, we need to explore different domains for more variants of this relationship between processes and events.

Conclusion

This paper has approached the problem of capturing organizational dynamics by addressing the representational problems involving occurrents (states, events and processes). The temporal nature of events and processes in this domain is elucidated. The causation relation between processes and events have been formalized. Events in this domain are instantaneous while processes take place over intervals. A process causes a transition event to happen and that transition event is deemed to have happened instantaneously.

References

[1] Galton, A. 2010. “Spatial and Temporal Knowledge Representation” Third Indian School on Logic and its Applications 18-29 January 2010.

[2] Gruber, T.R. 1993. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5 (2): 199-220.

[3] Minz, S. 2006. ”Ontologies and Much More” PowerPoint Presentation.

[4] Rosengarg, J. 2004. “Ontological Representations of Software Patterns”.

[5] Grenon, P. and Smith, B. 2004. “SNAP and SPAN: Towards Dynamic Spatial Ontology”, Spatial Cognition and Computation, 4:1, 69-103.

[6] Corazzon, R. 2010. “Theory and History of ontology”.

[7] Leinfellner, W., Kraemer, E. and Schank, J. 1981. “Language and Ontology”: Proceedings of the 6th International Wittgenstein Symposium, Kirchberg am Wechsel, Austria, 23 – 30 August, 1981.

[8] Guarino, N. 1998. Formal Ontology in Information Systems. Proceedings of FOIS’98, Trento, Italy, June 6-8, 1998. IOS Press, Amsterdam, pp. 3-15.

[9] Galton, A. 2006. (Power Point Presentation) “On What Goes On: The Ontology of Processes and Events” University of Exeter, UK.

[10] Jurisica Igor, Mylopoulos John and Yu Eric. 2004. “Ontologies For knowledge Management: An Information Systems Perspective”.

[11] Hayes, P.J. 1985. Naive Physics I: Ontology for Liquids. In Hobbs and Moore, eds., 71-107.

[12] Harel David. 1987. The Statemate Semantics of Statecharts.

[13] Allen, J.F. 1984. Towards a General Theory of Action and Time. *Artificial Intelligence* 23: 123-154.

[14] Grenon Pierre. 2010. “Spatio-Temporality In Basic Formal Ontology”

[15] Grenon, P. and Smith, B. 2004. SNAP and SPAN: “Prolegomenon to Geodynamic Ontology 1”.

[16] Yang, J. S. H., Chin, Y. H. and Chung, C. G. 1989. “Many-sorted First-Order Logic Database Language.

[17] Fox, M.S. and Gruninger, M. 1998. "Enterprise Modelling", AI Magazine, AAAI Press, Fall 1998, pp. 109-121.

[18] Allen, J.F. 1983. Maintaining knowledge about temporal intervals. Communications of the ACM 26 (11): 832-843.

[19] McDermott, D.V. 1982. A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science* 6: 101-155.

[20] Bacchus, F., Tenenberg, J. and Koomen, J.A. 1991. A non-reified temporal logic. *Artificial Intelligence* 52: 87-108.

[21] Haugh, B.A. 1987. “Non-Standard Semantics for the Method of Temporal Arguments”. Proceedings of 10th IJCAI 1: 449-455.

[22] Galton, A. 2004. “Operators Vs Arguments: The Ins and Outs of Reification”.

[23] Galton, A. 2012. States, Processes and Events, and the Ontology of Causal Relations (FOIS 2012).

[24] Allen, J. and Hayes, P. 1985. A common-sense theory of time. Proceedings of IJCAI’95: 528-531.

[25] Guarino, N. and Welty, C. 2000. “A Formal Ontology of Properties”. (LADSEB-CNR Padova, Italy, 2000).