An Upper Level Ontological Model for Engineering Design Performance Domain

Vadim Ermolayev[1], Natalya Keberle[1], and Wolf-Ekkehard Matzke[2]

[1] Department of IT, Zaporozhye National University, Zhukovskogo 66, 69063,
Zaporozhye, Ukraine
vadim@ermolayev.com, kenga@zsu.zp.ua
[2] Cadence Design Systems, GmbH, Mozartstr. 2 D-85622 Feldkirchen, Germany
wolf@cadence.com

Abstract. The paper presents our upper level lightweight descriptive model for the set of the Core ontologies of PSI[1] Suite. While PSI Suite of Ontologies is an interlinked modular library of ontologies describing the domain of engineering design performance in microelectronics, PSI upper level ontology is more domain-independent. It is a model of stateful creative dynamic processes, proactive agents, and objects situated in nested dynamic environments based on formal representation of time, events, and happenings. It may be used as an upper level theory for domain ontologies in different application domains having common features. PSI upper level ontology is designed as a semantic bridge facilitating to mapping PSI Domain ontologies to abstract ontological foundations and common sense. It is also used as semantic "glue" for bridging PSI ontologies with other theories, widely accepted in the domains where processes, states, and participating objects are the major entities. These mappings and semantic bridges are supposed to ease the commitment of potential users to PSI Suite. PSI upper level ontology is also used as a "proxy" for different kinds of evaluation of PSI ontologies in frame of our "shaker modeling" methodology for ontology refinement.

1 Introduction

PSI project develops the methodology and the toolset for assessing, predicting, and optimizing the performance of engineering design systems in microelectronics. Though design technology in this domain is well defined, many factors make design processes highly stochastic, non-deterministic, structurally ramified, time-bound – in a phrase, loosely defined and highly dynamic. The examples of such factors are: human factor, innovative character, the pace of technology change, the peculiarities of the market and customer requirements, etc. PSI uses simulation to observe and predict the course of a Dynamic Engineering Design Process (DEDP) with sufficient detail for making assessments grounded. Simulation allows playing "what-if" games to model the unpredictable character of the real business world of microelectronic design.

[1] Performance Simulation Initiative (PSI) is a research and development project of Cadence Design Systems, GmbH.

Q. Li et al. (Eds.): ER 2008, LNCS 5231, pp. 98–113, 2008.

Finely grained and explicit knowledge of design processes and environments is an important intellectual asset which allows PSI methodology be convincing. This knowledge is formalized using the Suite of PSI ontologies. If someone imagines an arbitrary design flow, most certainly he or she will think in terms of: a goal – the state of affairs to be reached; an action which may bring the process closer to its goal; an object to apply actions to; a designer who acts and applies actions to objects; an instrument to be used by an actor to execute actions; and an environment in which the process occurs. All these interact in dynamics – depending on time and on events which manifest the changes in a design system which is the environment of a DEDP. The structure of the Suite of PSI Ontologies reflects this approach. It comprises six cross-linked Core ontologies: Time ontology; Environment, Event and Happening ontology; Actor ontology; Project ontology; Process and Process Pattern ontologies; and Design Artifact ontology. The "corolla" of this Core is formed by Extension ontologies collaboratively developed in PSI and PRODUKTIV+[2] projects. The most important Extensions are: Resource ontology with Tool package, Generic Negotiation ontology.

The ontology presented in this paper is the upper level part of PSI Suite of Ontologies. Its main purpose is putting the components of the Suite in line with the commonly accepted metaphysical and cognitive framework of the common sense represented by chosen reference ontologies. Additionally, we aim at providing semantic bridge to mainstream enterprise, business, and process modeling frameworks. Bridging PSI ontologies to these mainstream theories of process knowledge representation facilitates to easier commitment of engineering design domain professionals to the Suite. Upper level ontology also plays an integration and harmonization role in PSI Suite because it represents a rather domain-independent descriptive theory based on formal principles for harmonizing and integrating the underlying domain dependent modules with other relevant ontologies. In addition to being the semantic "glue" between the Suite and the outer world of knowledge representation presented ontology plays an important role in the methodology of knowledge engineering in PSI. It is the resource which is intensively used in the refinement and the evaluation of PSI Core Ontologies.

The rest of the paper is structured as follows. Section 2 outlines requirements and objectives which shaped out our upper level model. Section 3 puts PSI upper level ontology in the context of related work and presents our ontological choices. Section 4 outlines the taxonomy PSI upper level ontology and discusses the semantic contexts of its key concepts in detail. Section 5 sketches our ontology engineering methodology. Section 6 reports on the implementation and evaluation of the described ontology. Finally, concluding remarks are given and our plans for future work are outlined in Section 7.

2 Modeling Requirements

PSI project aims at developing models, methodologies, and software tools providing for rigorous engineering treatment of performance and performance management. PSI performance modeling and management approach focuses on performance as a pro-active action. A fine-grained dynamic model of a DEDP and a design system is

[2] PRODUKTIV+ is the R&D project funded by the German Bundesministerium für Bildung und Forschung (BMBF).

therefore developed. PSI approach considers that performance is embodied in its environment and is controlled by the associated performance management process.

A DEDP is a goal-directed process of transforming the representations of a design artifact in stateful nested environments. An environment comprises design artifact representations, resources, tools, and actors who perform actions to transform design artifacts using tools, consume resources. Actions are admissible in particular environment states and may be atomic or compound, state-transitive or iterative, dependent or independent on other actions. The components of an environment may generate internal events or may be influenced by external events. Events may have causal dependencies. A DEDP is a problem solving process which goals, partial goals, and environments may change dynamically. A decision taking procedure is associated with each state to allow environments adjust the process taking these changes into account. Decisions are taken by actors modeled by software agents.

PSI software tools are developed [1] for assisting project managers to make robust planning, monitoring, and management of their design projects aiming at reaching best possible performance. Grounded decisions in planning are based on the knowledge base of project logs accomplished in the past. These logs provide vast and finely grained records of the performance of accomplished projects and may be used for simulating the behavior of the design system in response to different influences. At project execution phase PSI software may be used for predicting the behavior of the design system in the future based on the record of the partially accomplished DEDP, the knowledge about its environment(s), and performance simulations.

Mentioned functionalities may only be implemented if a rich and expressive domain model is used. This model should be capable of facilitating agents reasoning about environments, events, and actions employed in decision taking procedures enacted at environmental states. These sorts of commonsense reasoning require ontological representations of time [2], environments, events and their subjective perceptions [3], processes, actions, actors, design artifacts, resources, tools [4, 5]. The models of these domain aspects form the Core and the Extensions of PSI Suite of Ontologies v.2.1 and v.2.2 [5].

3 Related Work and Modeling Choices

The main function of our upper level ontological model is putting the components of the Suite in line with the commonly accepted metaphysical and cognitive framework of the common sense. Common sense knowledge is captured by several foundational ontologies. Important examples are Suggested Upper Merged Ontology (SUMO) [6], Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [7], the upper level of OpenCYC [22], Basic Formal Ontology (BFO) [7]. Highly reputable linguistic resources like WordNet Linguistic Ontology (WordNet) [8] should also be considered. One more objective of introducing the upper level of the Suite is providing semantic bridges to mainstream enterprise, business, and process models like the Enterprise Ontology (EO) [9], Toronto Virtual Enterprise Ontology (TOVE) [10], Process Specification Language (PSL) [11].

In difference to the mentioned enterprise, business, and process modeling frameworks, which are, to a certain extent, domain independent (TOVE, PSL) or model manufacturing Domain (EO), PSI upper level ontology defines an abstract descriptive theory for the domain of engineering design processes and environments. As many foundational ontologies PSI upper level ontology has a clear cognitive orientation in the sense that it does not pretend being strictly and rigorously referential to the theories describing nature. Instead, it captures ontological categories and contexts based on human common sense reflecting socially dominant views on the Domain – characteristic at least to engineering design professionals. As such, the categories introduced in our ontology are not related to the intrinsic nature of the world but are rather thought of as "cognitive artifacts ultimately depending on human perception, cultural imprints and social conventions" [7]. Therefore, these categories assist in making already formed conceptualizations of PSI Suite of Ontologies explicit and referenced by the common sense. PSI upper level ontology also plays an integration and harmonization role of a foundational ontology [12] because it represents a rather domain-independent descriptive theory based on formal principles for harmonizing and integrating the underlying domain dependent modules with other relevant ontologies.

In contrast to foundational ontologies PSI upper level ontology is not foundational in the sense that it is not a profound and a complete theory in philosophical or, more precisely, cognitivistic sense. For example, it does not deal with many problems characteristic for foundational theories like: differences between abstract and concrete objects, particulars and universals; spatio-temporal co-localization of things; mereological axiomatization; etc. It also does not provide rich axiomatic sets for rigorously describing the semantics of the contained entities. Instead, other highly reputable foundational ontologies are used as reference sources for defining PSI upper level ontology components. The mappings of these components to those reference sources are explicitly specified. Choosing the most appropriate reference foundational ontologies among possible candidates is not an easy task because it requires ontological commitment to the chosen ontologies and their ontological choices. Typical ontological choices (also called meta-criteria) are: (i) Descriptivism vs. Revisionarism; (ii) Multiplicativism vs. Reductionism; (ii) Possibilism vs. Actualism; (iv) Endurantism vs. Perdurantism. A good comparative analysis of several well known foundational ontologies and their ontological choices has been undertaken in SmartWeb project [13]. Five most promising candidates among approximately a dozen available worldwide has been analyzed: BFO, DOLCE, Object-Centered High-level Reference Ontology (OCHRE) [7], OpenCYC, and SUMO. The results are given in Table 1. Typical ontological choices in line with modeling requirements of PSI project are discussed below. Based on this discussion our choice of reference foundational ontologies for the design of our upper level ontology is made.

Table 1. Foundational ontologies and their ontological choices [13]

Ontology **Alternative**	BFO	DOLCE	OCHRE	OpenCYC	SUMO
Descriptivism	-	+	-	+	+
Multiplicativism	-	+	unclear	unclear	+
Actualism	+	-	-	unclear	unclear
Perdurantism	+	+	-	unclear	+

Legend: + – the ontology supports the ontological choice; - – the ontology does not support the ontological choice; unclear – it is not clear if the ontology supports the ontological choice.

Descriptivism vs. Revisionarism. A descriptive ontology aims at describing the ontological assumptions based on the surface structure of natural language and human common sense. For example, a descriptive ontology distinguishes between physical and abstract objects based on the human common sense perception of these categories. It is common to consider that a physical object is a category of things which have tangible physical properties, can be sensed, are extended in space and time. On the contrary, an abstract object does not possess the abovementioned properties. A revisionary ontology is committed to capture the intrinsic nature of the world. As a consequence, such a commitment may impose that only entities extended in space and time exist. Though we refrain from modeling abstract things in PSI as much as possible[3], we still have to model immaterial things which are not made of matter, do not possess spatial properties, etc. Therefore, revisionarism would have been a wrong choice for PSI. PSI upper level ontology is a descriptive ontology and has to be based on a descriptive foundational ontology like DOLCE, OpenCYC, or SUMO.

Multiplicativism vs. Reductionism. A multiplicative ontology allows different entities to be co-localized in the same space-time. The difference of these entities means that they have different essential properties. For example, a silicon wafer of a chip (a material object) and a definite amount of silicon this wafer is made of (an amount of matter) are co-localized in space-time for the whole life of this particular chip. Reductionistic ontology postulates that each space-time location contains at most one object. Differences in essential properties are regarded as being linked to different points of view from which one can look at the same spatio-temporal entity. Reductionistic approach therefore extracts all essential properties different from spatio-temporal ones from entities and places them to the views on these entities. In PSI it is considered that an entity possesses all its essential properties and the views on an entity may reveal different subsets of these properties depending on the point of view. For example, an agent may be (i) a model of one physical person – a designer; (ii) a model of a group of designers working on one design project – a development team. PSI upper level ontology should therefore be a multiplicative ontology – like DOLCE or SUMO.

Possibilism vs. Actualism. An actualistic ontology postulates that everything that exists is actual. Things that are not actual and, therefore, do not exist may be withdrawn from consideration. Different forms of possibilism are based on different ways of the denial of this postulate. For example our beliefs, which are hypotheses based on incomplete, partial knowledge about the world, are very often roughly equally believed possible alternatives. Considering such alternatives is characteristic to human common sense and cognition. Committing to possibilism means being able to represent possibilia – possible alternative entities in a domain corresponding to different modalities in different possible worlds. Possibilism is particularly useful in reasoning about future courses of processes and about actions [14]. PSI upper level ontology has to be capable of modeling possibilia. For example, a design process depending on the future events in its environment may take one of the possible alternative courses. These alternative courses should all be considered and analyzed for choosing the best possible one to follow. Hence, we have to commit to possibilism of a foundational ontology like DOLCE or OCHRE.

[3] All concepts of PSI upper-level ontology are not abstract – Fig. 1.

Endurantism vs. Perdurantism. A fundamental ontological choice is the commitment to a way of modeling changes of things in time. Endurantism (also called 3D paradigm) postulates that all things do not change in time in the sense that all the proper parts of an entity (a whole) are present in this whole at any moment of the existence of this whole. Differently to that, perdurantism (also called 4D paradigm) assumes that entities may have different parts at different moments of their existence – meaning that entities have both spatial and temporal constituents. PSI upper level ontology needs to model both endurants and perdurants. Indeed, many of the concepts characteristic to engineering design always contain all their parts, but many other of them are composed of temporally different parts – like phases in a design process. Therefore, a reference foundational ontology for our ontology should be based on 4D paradigm, comprising 3D as a particular case. Such ontologies are BFO, DOLCE, and SUMO.

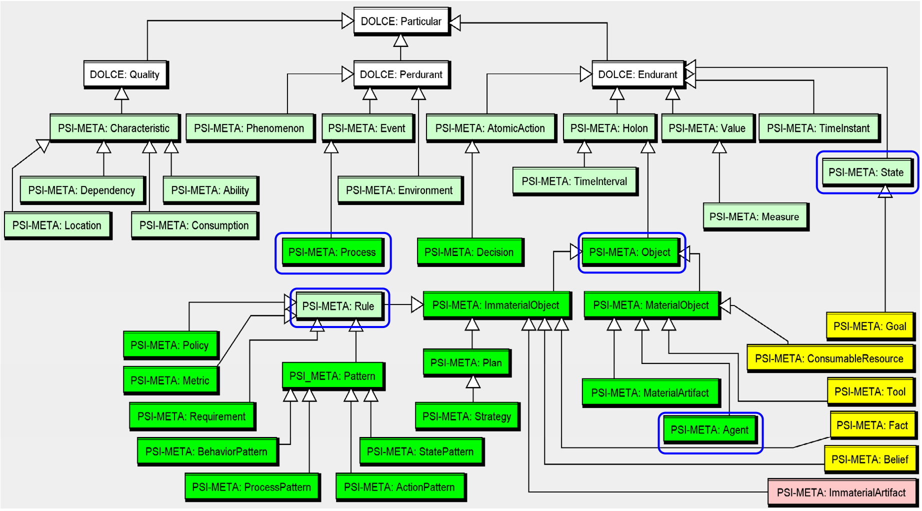
Hence, only DOLCE commits fully to all the ontological choices required by PSI. SUMO provides all the necessary features except possibilism. This is why we use the upper taxonomical level of DOLCE as our foundational framework. We also use SUMO extended by WordNet as a target for mapping the concepts of our upper level ontology and PSI Core Ontologies.

4 PSI Upper Level Ontology

The postulates, assumptions, objectives, and ontological choices of PSI modeling approach were presented in Sections 2 and 3. Here the semantic contexts of several key concepts of PSI upper level ontology[4] are discussed in detail: a Process, a State, an Object, an Agent, and a Rule. Fig. 1 pictures the taxonomy of PSI upper level ontology.

A Process (Fig. 2a) is a specialization of an Event [3] which is stateful and possesses pro-active character. A Process has its Environment – the part of the world which may influence the course of the Process or may be changed in the course of the Process. A Process is pro-actively directed by the Agent who manages it. Proactiveness of the Agent is understood in the sense that the Agent pursues a particular Goal in the managed Process. This Goal is the State of the Environment which the Agent desires to make reached. It should also be mentioned that the change in the Environment is not produced by the Process, but by the entities who act in this process – those Agents who execute AtomicActions wrapped by the Process. In general, it is considered that changes may only be applied by Agents through execution of AtomicActions. For example, it is wrong to say that a multimedia controller layout has been designed by the process of logical design. In fact the appearance of the layout for the multimedia controller in a certain state of the Environment (the measurable change in the Environment [3]) has been achieved by the team of Agents who executed a particular sequence of AtomicActions. By that the Agents applied the sequence of particular changes to the Environment and guided the environment through the sequence of States towards the Goal. Processes in an engineering Environment

[4] Complete specification could be found at http://ermolayev.com/psi-public/PSI-META-v-2-2-Spec.pdf



Legend:
(i) Concepts are colored reflecting classification by OntoClean property types (Section 6):
□ - category, 🟩 - type, 🟩 - quasi-type, 🟨 - material role, 🟥 - mixin.
Categories, types, and quasi-types form the backbone taxonomy.
(ii) Semantic contexts of the concepts in rounded rectangles (▢) are discussed in detail.

Fig. 1. The taxonomy of PSI upper-level ontology

can not connect any arbitrary State to any other arbitrary State because it is senseless with respect to the technology or the methodology. Some sequences of States may therefore be withdrawn from the engineering design routine and some other sequences of States may be suggested or prescribed by an industrial standard or a company policy. These prescriptions in terms of PSI upper level ontology are ProcessPatterns.

Any Process, as a pro-active stateful manifestation of a change in the Environment, is guided by its managing Agent to reach the State (Fig. 2b) of affairs in which the constituents of the Environment possess the properties partially or fully matching the Goal of that Agent. It is considered that a Process has reached its target State if such a state of affairs is reached. Otherwise the Process fails to reach its target State. A Goal, if complex, can be decomposed to simpler partial Goals as often done in problem solving. Such partial Goals are in fact the states of affairs that should be reached before the overall compound Goal can be attacked. States in upper level ontology are the configurations of the constituents of an Environment. It is considered that a State is reached when the constituents of the Environment have properties with Values in the ranges satisfactory matching the corresponding Goal or partial-Goal[5] of an Agent. In engineering design mentioned Goals are technologically controlled. For example, a technology of digital front-end design in microelectronics and integrated circuits prescribes that an overall Goal of a digital back-end design is the development of a

[5] In our agent-based software implementation such decomposition is the substantial part of work breakdown structure generation [1]. Goal decomposition is based on the ontological representation of design task and activity patterns provided by the Process Pattern Ontology of the Core of the PSI Suite.

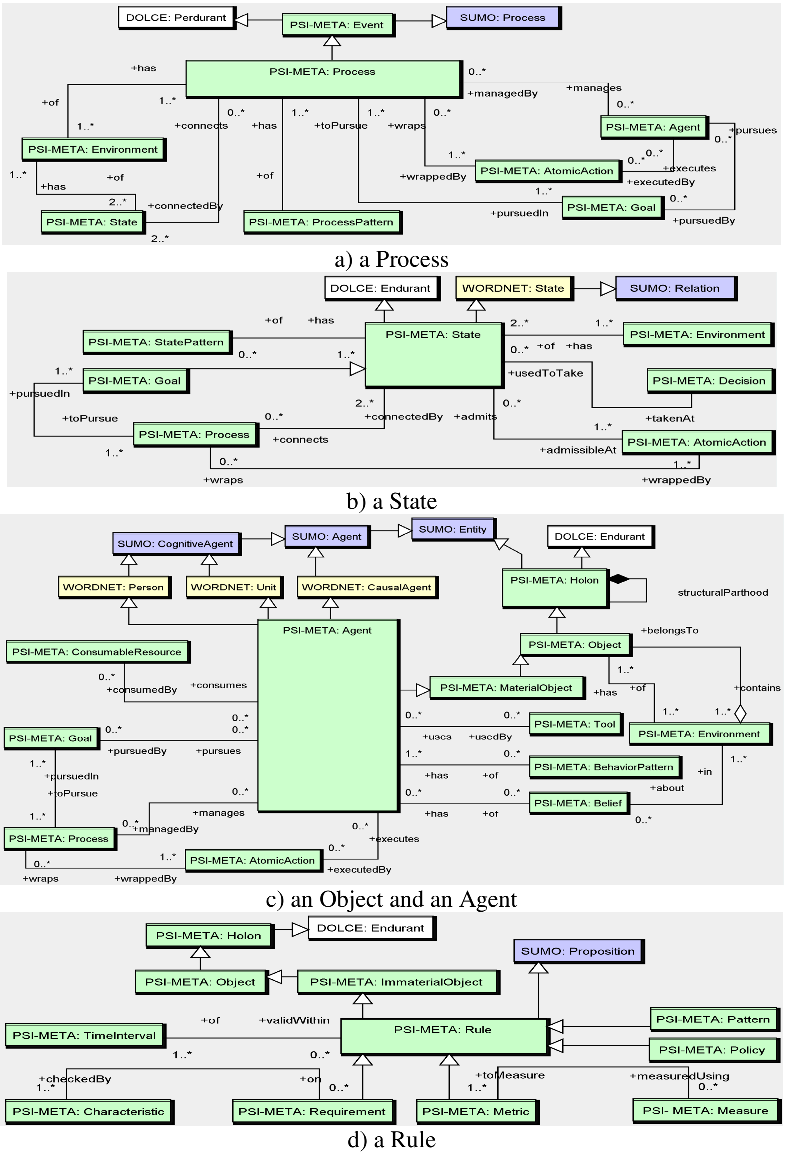


Fig. 2. Semantic contexts of the key concepts

design artifact in GDSII layout representation. At the same time the technology suggests that netlist, floorplan, placement and routing representations should be developed before the overall Goal can be reached. In these settings the States can be seen as technological milestones on the path through the problem solution space leading to the overall Goal. The requirements to the ranges of the property values of the constituents of the Environment are denoted by StatePatterns. StatePatterns are controlled by the Policies of a company which should be based on the standards of the particular industrial sector. Goals and corresponding partial Goals may be pursued by taking different alternative paths going through different States. If a problem solution space is represented as a directed graph, a State may have several alternative outgoing edges. These edges correspond to different admissible AtomicActions applying different changes to the Environment. A Decision on the choice of an admissible AtomicAction should be taken for choosing the continuation of the path at any State. In particular, a Decision in the target State chooses among the alternative to terminate the process in success and the alternative to refine the values of the properties of the constituents of the Environment heading to the same target State. Hence, a Decision is a specific AtomicAction which applies changes to an Environment indirectly – by choosing the alternative on the solution path. A Decision is also a mechanism to alter the course of the Process when the Goal or the sub-Goals are dynamically changed. In difference to an Environment, which is a Perdurant, a State is an Endurant because all its parts should be present at any TimeInstant of the presence of a State. A State is not a Holon because a State can not be a part of another State.

An Object (Fig. 2c) is a Holon which has Environment, belongs to an Environment, and may be changed in the course of the execution of an AtomicAction. An Object may have Characteristics, though the relationship of an Object to a Characteristic is not explicitly specified at this level of abstraction. The reason is that we refrain from letting the subclasses of an Object inherit this very generic relationship. Instead we prefer to specify individual relationships between the subclasses of an Object and a Characteristic at lower abstraction levels – for example in the Core or the Extensions of PSI Suite. An Object could be either material or immaterial. MaterialObjects are those Objects which are physically or legally substantial in the sense that they possess tangible physical non-temporal properties like mass, color, shape, size, speed, usage right and can not be copied or duplicated without borrowing a definite amount of physical or legal[6] substance for it. The law of conservation of matter is applicable to material objects. MaterialObject subclasses are an Agent, a MaterialArtifact, a ConsumableResource, a Tool. ImmaterialObjects in contract to material ones are not substantial in physical or legal sense. Hence, they can be copied or duplicated without consuming physical or legal substance for that. ImmaterialObject subclasses are an ImmaterialArtifact, a Rule, a Plan, a Fact, a Belief.

An Agent (Fig. 2c) is a MaterialObject who possesses pro-activity, is able to execute AtomicActions and manage Processes. Pro-activity of an Agent is revealed in pursuing Goals of changing the Environment to a desired State. An Agent is the only entity which can change its Environment by executing AtomicActions applied to the Objects in the Environment. An Agent has Beliefs about its Environment(s) which are the hypotheses believed to be true. These Beliefs may further become Facts if confirmed by the happenings [3] perceived by the Agents. Beliefs together with desires and intentions are important basic elements forming the behavior of an Agent. This behavior is regulated by BehaviorPatterns specified as Rules. An Agent is an abstract entity which is a generic model for an individual person (a manager, a designer), a

[6] By an odd term of "legal substance" we mean a legal permission to have an extra copy of an Object which is not a physical object in the sense of SUMO or DOLCE. A good example of such an Object is a software program with a license (legal substance).

group of persons or artificial agents acting on behalf of physical persons (a team or an organizational unit), or an external pro-active entity influencing the Environment of an observed Process in a definite way. These aspects of an Agent are specialized and refined at the lower abstraction levels by PSI Core ontologies.

A Rule (Fig. 2d) is an ImmaterialObject which is a principle, a condition, a procedure, a generic pattern, or a norm regulating possible process, action, behavior, or state of affairs. As far as a Rule subsumes to a Holon it inherits structural parthood relationship of a Holon. Hence, a Rule may be an atomic proposition or a more complex composition of other Rules. As far as a Rule is an Endurant no temporal parthood relationships are allowed for its proper parts – the composition of a rule can not be changed in time. A Rule itself still has a temporal property of validity – it is valid within a particular TimeInterval or several particular TimeIntervals. If a Rule is a principle or condition that customarily governs behavior then it subsumes to WordNet: Rule and further on to SUMO: Proposition. If a Rule is a generalization that describes recurring facts or events then it subsumes to WordNet: Law and further on to SUMO: Proposition. If a Rule is something regarded as a norm constraining possible action or behavior then it subsumes to WordNet: Regulation and further on to SUMO: Proposition.

5 Ontology Engineering Methodology

The methodology used in the design of PSI upper level ontology may be identified as "shaker modeling". It is the combination of bottom-up and top-down modeling techniques exercised in subsequent design iterations. The source for the top-down activity is PSI Theoretical Framework. The source for the bottom-up phase is domain knowledge acquired from subject experts and formalized in the Core of PSI Suite of Ontologies. Both kinds of sources are refined in iterations before performing the phases of upper level ontology design. The sources are also aligned to the foundational reference ontology and mapped to the common sense reference ontology using the upper level ontology as a semantic "glue" in the last two phases of every design iteration.

The most recent revision of the Theoretical Framework is used in the first phase – skeleton design of the upper level ontology. The outcome is represented as a UML class diagram. Skeleton design phase is a top-down activity because a more abstract theoretical framework is used as a source for the development of a more elaborated descriptive theory. In the refinement phase the harmonization of the skeleton of the upper level ontology with the previous stable revision of the Core Ontologies of PSI Suite is performed. The objectives of this harmonization activity are: (i) ensuring that the upper level model does not contain components which contradict to the core-level model in their semantics and (ii) ensuring that all core-level concepts are properly mapped to the upper level concepts. Previous stable revision of PSI Core Ontolgies is used to ensure upward compatibility of the revisions of the PSI Suite comprising the upper level ontology. The outcome of this phase is presented in the form of three separate UML class diagrams: (i) the taxonomical structure of the upper level ontology; (ii) the diagram of the "horizontal" relationships among the concepts of the upper level ontology; (iii) the mappings of the concepts of the Core Ontologies to the concepts of the upper level ontology. Refinement phase is a bottom-up activity because an upper level model is harmonized with the lower-level one – the core part of the domain theory. At the beginning of the alignment phase upper level ontology is checked for the conformance to the ontological choices of the reference foundational ontology and its taxonomical structure is formally evaluated. As DOLCE is chosen as a reference foundational ontology for PSI, the upper level model is checked for being descriptive, possibilistic, multiplicative, and perdurantistic. OntoClean [15] is used as a methodology for formal evaluation of the taxonomy. As result, the taxonomy is refined and formally evaluated. Further on, upper level ontology concepts are mapped to the reference ontology which has been chosen as a source of common sense semantics – SUMO+WordNet in our case. These mappings allow checking if our upper level theory is sound enough to adequately conform to human beliefs about what the world is. If the result of such verification is positive (all the mappings are easily built and their semantics is easily understood), then we may expect that PSI upper level ontology will be accepted by humans without major difficulties. These common sense mappings may also be used as "referees" at the subsequent bridging phase. Bridging phase is actually not the phase of ontology design. It is the activity in which upper level ontology is used as a semantic bridge to help evaluating the Core of PSI Suite against the other ontologies describing similar domain theories.

The iterations of PSI upper level ontology development are organized as shown in Fig. 3. The whole process is performed in two stages: (i) initial design and (ii) iterative refinement similarly to what is suggested by DILIGENT [16] methodology of collaborative ontology engineering. It may be stated that DILIGENT in our approach is used as the higher-level methodological framework organizing iterations in a needed way. DILIGENT is used because the development of our Suite of Ontologies is done in a distributed dynamic environment (several local groups of subject experts from different organizations in frame of PSI and PRODUKTIV+ projects take part).

The stage of the initial design is the preparatory activity at the very beginning of ontology design process. Its objective is to develop the initial revisions of the Theoretical Framework and the Core set of the domain ontologies. An initial revision of the upper level ontology is developed at the end of the initial design stage because it requires both as sources. Two revisions of the Theoretical Framework have been developed before starting the design of the upper level ontology. The second revision is the result of the refinement based on the user evaluation feedback on the first revision of the Core ontologies. Hence, even the first revision of the upper level ontology is designed with the account for the user evaluation of the domain theory. An iteration of the refinement stage also uses the latest revision of the Theoretical Framework developed in this iteration and the revision of the Core ontologies built in the previous iteration. Iteration starts with the development of the Core set of ontologies based on the upper level ontology revision of the previous iteration and ends by the development of the new revision of the upper level ontology.

Several kinds of ontology evaluation activities are undertaken in each design iteration (Fig. 3). The first one is user evaluation. The objective of the user evaluation is to find out if the Core set of ontologies fits the requirements of user teams and the requirements of the software development based on this set of ontologies. An external evaluation by independent experts may also be done at this stage to ensure that evaluation results are unbiased and of good quality. It has been found out [17] that for PSI Core ontologies probably the best fitting methodology is Pinto and Martens [18].

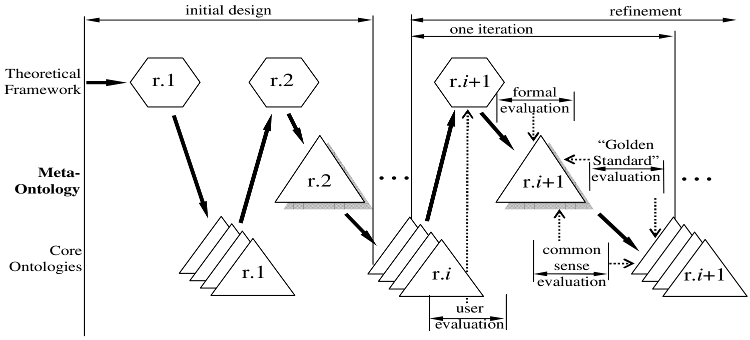


Fig. 3. Iterations of PSI upper level ontology design

The feedback of this iteration is taken into account in the refinement of the Theoretical Framework. The second kind of evaluation activity – formal evaluation, is undertaken at the Alignment phase after the newly developed revision of the upper level ontology has reached release candidate state. The objective of formal evaluation is to check the conformance of the taxonomy structure of the upper level ontology to meta-properties of rigidity, identity, and dependence [15]. The methodology for this kind of evaluation is OntoClean [15]. The results of the formal evaluation are used for the refinement of the release candidate of the upper level ontology. The third kind of evaluation activity is the evaluation of the Core set of ontologies versus the upper level ontology. Similarly to the formal evaluation it is performed at the Alignment phase. The conformance of the Core set to the common sense is now checked. The mappings of the Core ontologies to the reference common sense ontology (SUMO) are elaborated using the upper level ontology as the "glue", like for example in [19]. The result allows estimating how easily domain experts may (potentially) commit to the Suite of Ontologies. If the Core set maps well to the common sense reference ontology one may expect that the commitment of domain experts to it may be reached considerably easily. If the mapping is bad then the ontology is either a novel extension of the common sense conceptualization or, more probably, is badly designed. The feedback of this kind of evaluation is used in refining the Core ontologies and, later on, in refining the next revision of the upper level ontology. Finally, the fourth kind of evaluation is the comparison of the Core set of ontologies with the so called "Golden Standard" [20]. By "Golden Standard" we mean a highly reputable ontology describing the theory of the same or a similar domain which has already gained broad commitment by domain experts. The evidence of such a commitment may be that a "Golden Standard" ontology is the basics for a standard, a de-facto standard, or a standardization proposal. This kind of evaluation is performed at the Bridging phase. Similarly to the common sense evaluation the mappings of the Core set of ontologies to a "Golden Standard" are built. However, the objective of the evaluation is different. Completeness and expressiveness of the Core Ontologies are checked at this time. If all the concepts of a "Golden Standard" ontology are mapped by the concepts of the Core set then it may be estimated that the Core set covers the domain equally to or better than a "Golden Standard". Otherwise, the core set is less complete than the "Golden Standard". In the latter case the reasons of potential incompleteness should be analyzed. In a safe case it may be found out that the domain described by the "Golden Standard", though similar to ours, is broader. Otherwise, the Core set is incomplete. The mappings in the opposite direction – from the concepts of a "Golden Standard" to the concepts of evaluated Core ontologies, may help assessing the level of the expressiveness of the target. For example, if all the concepts of the "Golden Standard" map to single concepts of the evaluated Core set then it may be the case that the Core set possesses at least the same level of expressiveness at the "Golden Standard".

6 Ontology Implementation and Evaluation

PSI upper level ontology v.2.2 has been implemented in OWL-DL[7]. PSI Theoretical Framework v.2.0 [14] and Core ontologies of PSI Suite of Ontologies v.2.1 have been used as the knowledge sources. PSI Suite of Ontologies v.2.2 has been developed based on the elaborated upper level ontology v.2.2. Two different kinds of evaluation have been accomplished for the upper level ontology so far: formal evaluation and commonsense evaluation. Besides that, user evaluation of the set of the Core ontologies v.2.1 has been done before the beginning of the development of the upper level ontology v.2.2, as described in Section 5. "Golden Standard" evaluation is still in progress. User evaluation of the Core ontologies v.2.1 has been performed by the group of PSI software developers who used a goal-based evaluation routine to assess the appropriateness, the completeness, and the upward compatibility of the Suite of Ontologies. Appropriateness has been evaluated by checking if the Suite fulfils the requirements imposed by developed software. Completeness and upward compatibility with the previous revision has been checked by transferring the instances of the PSI knowledge base v.2.0 to v.2.1. User evaluation revealed minor problems which have been immediately resolved allowing us to fix v.2.1. Several issues have been listed as the ones for the future development. These issues have been taken into account in the revision of the Theoretical Framework v.2.0.

Formal evaluation of the taxonomy of PSI upper level ontology has been performed using OntoClean methodology [15]. The goal of taxonomy analysis is to verify if the structure of the taxonomy is formally correct. Other outcomes of this formal analysis are: (i) classifying taxonomy nodes according to OntoClean ontology of property types [21]; (ii) extracting the part of the analyzed taxonomy which is its backbone taxonomy [21]. In the course of this evaluation OntoClean meta-properties have been assigned to the concepts of the upper level ontology. After that OntoClean constraints have been applied to analyze if there are constraint violations in the taxonomy[8]. Applying OntoClean constraints to PSI upper level ontology subsumptions revealed no violations. Hence, the structure of the taxonomy is formally correct. Following [21], the concepts of PSI upper level ontology were classified according to

[7] Web Ontology Language, http://www.w3.org/TR/owl-guide/. OWL-DL implementation of PSI upper-level ontology is available at http://ermolayev.com/psi-public/psi-meta-v-2-2-draft.owl and its specification is at http://ermolayev.com/psi-public/PSI-META-v-2-2-Spec.pdf

[8] Detailed description of the results of this formal evaluation is given in PSI upper ontology specification.

OntoClean ontology of property types. As it has been found out, all its own concepts are Sortals. Non-sortal concepts are imported from DOLCE and are the categories forming the most upper part of the taxonomy. Among the sortals 16 are types and 17 are quasi-types. Categories, types and quasi-types form the backbone taxonomy of PSI upper level ontology. Among the remaining 6 concepts 5 are material roles and only 1 is a mixin. PSI upper level ontology does not contain phased sortals, formal roles and attributions. The backbone taxonomy and the parts of the ontology extending the backbone taxonomy are pictured in Fig. 1 using different shades of gray.

The objective of commonsense evaluation was to find out if the upper level ontology facilitates in mapping the Core ontologies to the reference foundational ontology. The mappings of the concepts of six PSI Core ontologies to WordNet+SUMO through PSI upper level ontology have been done using subsumptions. It has been found out that using upper level ontology as semantic "glue" made these mappings more precise and facilitated to defining the semantics of the concepts of the Core more explicitly. For example, looking up for a Project (the concept of PSI Project ontology) in WordNet+SUMO[9] reveals that a project is both: (i) "any piece of work that is undertaken or attempted" which subsumes to SUMO: IntentionalProcess and further to SUMO: Process; and (ii) "a planned undertaking" which subsumes to SUMO: Plan. The semantics of PROJECT: Project as specified in [5] reveals that a Project subsumes to PSI-META: Plan and consequently to SUMO: Plan. The mapping to SUMO: Process is therefore discarded as irrelevant. From the other hand, the analysis of the "hanging" concepts in the upper level ontology helps revealing the contexts in the Core which are still under-developed. For example, PSI-META: Goal does not subsume the concepts of the Core v.2.2. Therefore we may suspect that the aspects of goal-directed behavior in the Actor and Process Core ontologies v.2.2 still require refinement.

7 Concluding Remarks and Outlook

Presented ontology is the upper level descriptive theory for the Core set of PSI Suite of Ontologies. PSI Suite is an interlinked modular library of ontologies describing the domain of engineering design performance in microelectronics. PSI upper level ontology is more domain-independent. It formalizes an abstract theory of stateful creative dynamic processes, pro-active agents, and objects situated in nested dynamic environments based on the formal representation of time, events, and happenings. This upper level theory may be used as a higher-level framework for domain ontologies in different application domains having common features. PSI upper level ontology is designed as a semantic bridge formalizing the mappings of PSI Domain ontologies to abstract ontological foundations and common sense. It is also used as semantic "glue" for bridging PSI domain theory with other theories widely accepted in the domains where processes, states, and participating objects are the major entities. These mappings and semantic bridges are supposed to ease the commitment of potential users to PSI Suite. PSI upper level ontology is also used as a "proxy" for different kinds of evaluation of

[9] KSMSA Ontology Browser has been used: http://virtual.cvut.cz/ksmsaWeb/browser/title

PSI ontologies in frame of our "shaker modeling" methodology for ontology refinement. In its current revision presented ontology is still lightweight in the sense that it does not provide rich axiomatic definitions of domain semantics. The main reason is that in-depth domain axiomatization is done in the core of PSI Suite. However, some enrichment of the upper level theory with formal axioms for better describing bridges to common sense theories is planned for future work. One more direction of our future development is extending the sphere of influence of PSI upper level ontology to cover the Extensions of our Suite of Ontologies. We are also plan applying our upper level ontological framework in the domains adjacent to PSI and PRODUKTIV+. For example, it will be used in ACTIVE project[10] as one of the models for representing knowledge processes and their environments.

Bibliography

1. Sohnius, R., Jentzsch, E., Matzke, W.-E.: Holonic Simulation of a Design System for Performance Analysis. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) HoloMAS 2007. LNCS (LNAI), vol. 4659, pp. 447–454. Springer, Heidelberg (2007)
2. Ermolayev, V., Keberle, N., Matzke, W.-E., Sohnius, R.: Fuzzy Time Intervals for Simulating Actions. In: Kaschek, R., Kop, C., Steinberger, C., Fliedl, G. (eds.) UNISCON 2008. LNBIP, vol. 5, pp. 429–444. Springer, Heidelberg (2008)
3. Ermolayev, V., Keberle, N., Matzke, W.-E.: An Ontology of Environments, Events, and Happenings. In: Elci, A., Kone, M.T., Orgun, M.A. (eds.) Proc 3d Int. Workshop on Engineering Semantic Agent Systems. IEEE Computer Society CPS, Los Alamitos (2008)
4. Ermolayev, V., Jentzsch, E., Karsayev, O., Keberle, N., Matzke, W.-E., Samoylov, V., Sohnius, R.: An Agent-Oriented Model of a Dynamic Engineering Design Process. In: Kolp, M., Bresciani, P., Henderson-Sellers, B., Winikoff, M. (eds.) AOIS 2005. LNCS (LNAI), vol. 3529, pp. 168–183. Springer, Heidelberg (2006)
5. Ermolayev, V., Jentzsch, E., Keberle, N., Sohnius, R.: Performance Simulation Initiative. The Suite of Ontologies v.2.2. Reference Specification. Technical report PSI-ONTO-TR-2007-5, VCAD EMEA Cadence Design Systems, GmbH (2007)
6. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: Guarino, N., Smith, B., Welty, C. (eds.) Int. Conf. on Formal Ontologies in Inf. Systems, pp. 2–9. ACM Press, New York (2001)
7. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: WonderWeb Deliverable D18 Ontology Library (final), ICT Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web (2003)
8. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
9. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. The Knowledge Engineering Review 13(1), 31–89 (1998)
10. Grüninger, M., Atefy, K., Fox, M.S.: Ontologies to Support Process Integration in Enterprise Engineering. Computational & Mathematical Organization Theory 6(4), 381–394 (2000)
11. Bock, C., Grüninger, M.: PSL: A Semantic Domain for Flow Models. Software Systems Modeling 4(2), 209–231 (2005)

[10] ACTIVE: Knowledge Powered Enterprise (http://www.active-project.eu/) is an Integrated Project funded by Framework Program 7 of European Union.

12. Mika, P., Oberle, D., Gangemi, A., Sabou, M.: Foundations for Service Ontologies: Aligning OWL-S to DOLCE. In: 13th Int. Conf. on WWW, pp. 563–572. ACM Press, New York (2004)
13. Oberle, D., et al.: DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO). J. Web Semantics 5(3), 156–174 (2007)
14. Ermolayev, V., Jentzsch, E., Matzke, W.-E., Pěchouček, M., Sohnius, R.: Performance Simulation Initiative. Theoretical Framework v.2.0. Technical report PSI-TF-TR-2007-1, VCAD EMEA Cadence Design Systems, GmbH (2007)
15. Guarino, N., Welty, C.: Supporting Ontological Analysis of Taxonomic Relationships. Data and Knowledge Engineering 39(1), 51–74 (2001)
16. Vrandečić, D., Pinto, S., Tempich, C., Sure, Y.: The DILIGENT Knowledge Processes. J. of Knowledge Management 9(5), 85–96 (2005)
17. Simperl, E.: Evaluation of the PSI Ontology Library. Technical report, DERI Innsbruck, Austria (2007)
18. Pinto, H.S., Martins, J.P.: A Methodology for Ontology Integration. In: Gil, Y., Musen, M., Shavlik, J. (eds.) 1st Int. Conf. on Knowledge Capture, pp. 131–138. ACM Press, New York (2001)
19. Keberle, N., Ermolayev, V., Matzke, W.-E.: Evaluating PSI Ontologies by Mapping to the Common Sense. In: Mayr, H.C., Karagiannis, D. (eds.) 6th Int. Conf. Information Systems Technology and its Applications, Gesellschaft für Informatik, Bonn. GI LNI, vol. 107, pp. 91–104 (2007)
20. Brank, J., Grobelnik, M., Mladenić, D.: A Survey of Ontology Evaluation Techniques. In: Grobelnik, M., Mladenić, D. (eds.) SiKDD 2005, pp. 166–169 (2005)
21. Guarino, N., Welty, C.A.: A Formal Ontology of Properties. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 97–112. Springer, Heidelberg (2000)
22. Matuszek, C., Cabral, J., Witbrock, M., DeOliveira, J.: An Introduction to the Syntax and Content of Cyc. In: Proc. 2006 AAAI Spring Symp. on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Stanford, CA (2006)