ENGINEERING AN ONTOLOGY FOR AUTONOMOUS SYSTEMS
The OASys Ontology

Julita Bermejo-Alonso[1], Ricardo Sanz[2], Manuel Rodríguez[2] and Carlos Hernández[2]
[1] *Dpto. de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, Madrid, Spain*
[2] *Autonomous Systems Laboratory (ASLab), Universidad Politécnica de Madrid, Madrid, Spain*
jbermejo@ing.uc3m.es, {ricardo.sanz, manuel.rodriguezh, carlos.hernandez} @upm.es

Keywords: Ontology engineering, Domain ontology, Autonomous systems.

Abstract: This paper describes the development of an ontology for autonomous systems, as the initial stage of a research programme on autonomous systems' engineering within a model-based control approach. The ontology aims at providing a unified conceptual framework for the autonomous systems' stakeholders, from developers to software engineers. The modular ontology contains both generic and domain-specific concepts for autonomous systems description and engineering. The ontology serves as the basis in a methodology to obtain the autonomous system's conceptual models. The objective is to obtain and to use these models as main input for the autonomous system's model-based control system.

1 INTRODUCTION

Knowledge Engineering research has addressed the use and development of ontologies as a mean to improve knowledge processes. An ontology as a conceptualisation of a specification (Gruber, 1993), provides a solid basis to build knowledge bases for a greater functionality and shareness among users. Ontologies allow defining an abstract and simplified view of the concepts, their properties and their relationships within a domain of knowledge. Ontologies organise this knowledge in an appropriate structure, providing a representation vocabulary specialised for a domain. Ontologies formalise, structure and express the semantic content in the form of entities, their properties and their relationships, paying attention to the granularity of the ontological elements. On the other hand, ontologies are developed with a pragmatic focus, having in mind a context and an intended use for a particular domain, generally being developed following a design method or methodology.

Ontological Engineering refers to the different activities in the development process, the methodologies to support it, and the languages and tools used for the deployment of an ontology (Gómez Pérez et al., 2004b). This paper describes the ontological engineering of an ontology for autonomous systems that we have carried out. We have developed an ontology, OASys, as a conceptual framework and software support for the domain of autonomous systems. Our approach has been to develop an ontology to consider not only the description but also the engineering process of this kind of systems, as part of a long-time research programme on a universal technology for autonomous systems. Our goal is to include both generic knowledge on systems, as well as the domain-specific on autonomous systems, providing a common vocabulary for all the stakeholders. The underlying idea is that the ontology should express the concepts, consider the constraints or relationships in an explicit way under some ontological commitments but most importantly build the ontology to be readable by computers. This way the ontology would become an engineering artefact within a software process developed to define and to implement autonomous systems. Ontological domain models can drive typical development phases, such as requirements, design and implementation. The ontology so understood, is a mapping of the philosophical meaning of ontology into knowledge-based systems epistemology.

The paper is organised as follows. Section 2 reviews current research on engineering ontologies for the domains of autonomous systems, and software engineering. Section 3 summarises both the requirements we considered necessary in our ontology for the domain of autonomous systems, as well as the way they were deployed. Section 4 explains the design decisions made whilst developing the ontology. Next, Section 5 describes the actual ontology obtained, formalised using software engineering techniques. Finally, Section 6 draws some concluding remarks on the ontology development, and additional tasks to carry out to improve and refine it.

2 RELATED WORK

The ontology for autonomous systems (OASys) engineered addressed two different but interrelated domains. Firstly, the domain of autonomous systems. The ontology will be used to describe the autonomous system's structure, function, and behaviour. Secondly, the domain of software engineering. Our ontology will also describe the autonomous system's engineering process. As part of our research, we reviewed former attempts to develop an ontology in both domains.

Related to the domain of autonomous systems, ontologies have addressed different kinds of autonomous systems: mobile robots, agent-based applications, and autonomic sytems. For mobile robots, the ontologies have been used as a knowledge-representation mechanism to conceptualise their domain, their tasks or the environment where the mobile robots act. The research generally focuses on the description of the ontologies (Uschold et al., 2003), on their use for a particular mobile robot or application (Barbera et al., 2004), (Scrapper and Balakirsky, 2004), (Schlenoff and Messina, 2005), (Hallam and Bruynickx, 2006), and on the benefits achieved (Stojanovic et al., 2004b), (Provine et al., 2004). In the case of agent-based systems, the research on ontologies emphasises the necessity to share and to exchange knowledge among the agents in the autonomous system, and the problems of interoperatibility (Malucelli et al., 2005), (Schlenoff and Messina, 2005). For autonomic systems, ontologies have been developed to support information exchange and integration (Lehtihet et al., 2006), as part of the autonomous system (Tziallas and Theodoulidis, 2003), and as an explicit representation of data semantic and rules (Stojanovic et al., 2004a). In general, the research on ontologies for autonomous systems have focused on their usage, rather than providing a detailed account of the ontological engineering process that obtained the ontology

When it comes to the other domain of interest for our research, ontologies have been developed to act as domain ontologies to describe software engineering processes or technologies (Hesse, 2005), (Ruiz and Hilera, 2006), (Abran et al., 2006), (Hruby, 2005), (Falbo et al., 2005). Additionally, ontologies have been used as software elements within the system's architecture to support the software process (Eberhart, 2003), (Wongthongtham et al., 2005). The ontologies description has once again paid more attention to their benefits and use than to the specification, conceptualisation and formalisation of the ontological elements in the ontologies.

Our review pointed out the increasing use of ontologies for autonomous system's and for software engineering. Both domains have benefited from the use of ontologies, as they provide a common understanding of the concepts, allow sharing and transfering knowledge, and manage knowledge scalability. Nevertheless, the existing research did not provide enough elements to infer how the ontologies were engineered, in terms of their specification, conceptualisation and formalisation. These aspects are more commonly addressed as part of ontological engineering efforts (Uschold and King, 1995), (Noy and McGuinness, 2001), (Corcho et al., 2003), (Grüninger and Fox, 1995), without a specific domain such as the autonomous system's one under consideration.

Our approach to develop OASys combined the detailed description of the ontological engineering process as well as the analysis of the specific features to fulfill the requirements of the ontology to be used for the description and engineering of autonomous systems. Next sections describe all these aspects: the specification of OASys in terms of the requirements and the features to be accounted for; its conceptualisation as the design decisions we made; and the structure of the developed ontology taking into account the former elements.

3 OASys SPECIFICATION

A key aspect whilst developing an ontology is to state the ontology's purpose, which drives its development and ontological contents. Knowing what the ontology is to be developed for, allows focusing on the essential elements to be included. Additionally, it is necessary to define the type of ontology based on the subject of conceptualisation to consider. The level of abstraction, generality, and reusability of the ontological terms to be gathered in the ontology changes when considering an upper-level ontology from a domain one.

Different design criteria can serve as guideline to support the ontological engineering. These design principles allow evaluating the design with a focus on knowledge sharing (Gruber, 1993): clarity as the ontology is designed to communicate and to share the meaning of defined terms; extendibility to foresee the necessity to define or add new terms without modifying previous ones; minimal encoding bias to prevent the development being too based on the final implementation language; and minimal ontological commitments to allow instantiating the terms by the different users. Not all criteria can be met when designing an ontology. It is necessary to establish trade-offs between them and to compromise between the ontology design and its intended use.

The development of our ontology for autonomous systems took into account these design principles, prior to their implementation in the actual features of the ontology. The next sections summarise firstly the set of requirements considered in the development, and secondly how these aspects were finally addressed in the ontology.

3.1 OASys Requirements

- *Purpose*: the ontology would need to conceptualise the ontological elements to be used in the description of the autonomous system. Moreover, it aims at capturing the concepts required to define its generic engineering process. Our aim is to provide the system's developers with the ontological elements necessary to describe the autonomous system, from a more general viewpoint to a particular one. We are interested not only on the autonomous system, but also on the engineering development of this kind of systems as part of our research. These are the two aspects we are interested in as part of our research. We need to describe any autonomous system's structure, function and behaviour. We will also develop a methodology for autonomous system's engineering, hence the necessity to consider the engineering process.
- *Type of Ontology*: it would be a domain ontology to describe the autonomous system domain. A domain ontology provides the concepts and their relationships within a domain, about the activities in it and about the theories and principles guiding that domain. Being a domain ontology allows a high level of usability as it captures the domain knowledge in a problem-solving independent manner, being its reusability constraint to autonomous systems related aspects.
- *Design Criteria*: to assure the coherence and quality, the ontology would be developed bearing in mind the aforementioned design criteria. The ontology development has paid special attention to clarity, extendibility, minimal encoding bias, and minimal ontological commitment.
- *Knowledge Acquisition*: it would be made by considering different sources such as documents, existing ontologies, and experts. Documents such as articles, technical reports, or books serve as an input source for the ontological elements to be considered. Existing ontologies should also be reviewed, as the domain might have already been conceptualised, however with a different viewpoint or purpose. These existing ontologies should be selected, evaluated, and finally fully or partially reused, paying attention to the level of granularity (if the existing ontology covers the same level of detail as in the ontology under development). Domain experts also act as a possible source for the conceptualisation, since they provide their terminology, i.e., the words and terms in a domain they are familiar with.
- *Methodology*: the election of the methodology to follow during the ontology building is also an important factor to consider. There is a wide range of methodologies that have been developed to support and guide this process, as reviewed in (Gómez Pérez et al., 2004a). It would be necessary to assess them, to be reused in the ontology development. However, a new methodology can be defined or refined if existing ones do not fulfil the requirements for the development of a particular ontology.
- *Formalisation*: the ontology can be formalised using either traditional ontological languages or software engineering techniques. An analysis of the benefits and drawbacks of each option would be made to select the final formalisation technique.

3.2 OASys Features

Once the ontology requirements were established, we considered the actual ontology features and additional elements to fulfil each one of them. This section describes how the requirements were finally deployed in the ontological engineering process of OASys.

- *Structure*: the ontology needed to address two different aspects in its structure, the knowledge contents and the intended use. The knowledge contents refer to the type of ontology, considering different levels of abstraction to separate generic knowledge from domain-specific one. The intended use relates to the purpose of the ontology, as the distinction between the knowledge on autonomous system description and the knowledge about its engineering process.

  To address the different levels of abstraction in the ontology contents, the ontology has adopted a layered structure to address both generic and domain-specific knowledge. The upper layer contains the more abstract level knowledge, such as the concepts related to system's theories, and additional mereotopological concepts to express mereological and topological relationships in a system. A lower layer gathers the ontological elements to charaterise an autonomous' system structure, function and behaviour. The focus lies on the autonomous system domain as conceptualised in the framework of our research programme, however being general enough to be re-used in the development of any autonomous system.

To tackle the intended purpose for both the autonomous system's description and its engineering, we found a sensible idea to consider two ontologies as part of OASys: the ASys Ontology gathers the ontological elements to be used in the description of an autonomous system; the ASys Engineering Ontology collects the entities and relationships to describe and support the engineering process of an autonomous system.

- *Design Criteria*: the design criteria were followed throughout the development of the ontology. The original requirement regarding a design principle was analysed, and finally accomplished during the development of the ontology.
  - Clarity: to benefit from concept definitions which have already been used within the scientific community, existing ontologies and glossaries were reviewed. This aspect was specially important for the upper layer concepts, as they refer to generic knowledge on systems and systems' engineering. The concepts of the domain–specific layer were defined from our research outputs and other sources. The different available documents were carefully analysed to extract the ontological elements, checking for mismatches or commonalities. Those concepts would be later discussed with the group members to commit to the desired meaning for our research. Finally, all the ontological constructs (concepts, relationships, attributes, axioms) were defined in natural language.
  - Extendibility: to cater for extensions in the future, concepts were organised using subontologies and packages. Subontologies group ontological elements at the different abstraction levels for each one of the two developed ontologies. Packages are organisational elements to further classify the concepts within a subontology according to a concrete aspect. The ontology's scalability profits from these organising elements, allowing thus the extension or modification of the ontology without major changes to its structure and composition. Within a layer, new subontologies can be added to extend the existing ones with the purpose of addressing a new viewpoint when the ontology is to be extended. Within a subontology, new packages can be added or existing ones can be modified or updated by adding or removing concepts.
  - Minimal encoding bias: to prevent the incorrect conceptualisation of concepts based on the final implementation language syntax or appearance, intermediate tabular representations and graphs were used to define the different ontological constructs.
  - Minimal ontological commitment: only the fundamental concepts were described as agreed by the ontology users both at a generic knowledge, and at a domain–specific level. This latter elaborates the former by adding new ones to provide a deeper level of detail.
- *Inputs and Sources*: documents and existing ontologies were considered. Documents were analysed to come up with existing terminology and definitions for the different domains, subdomains, applications and aspects considered in the ontology's structure. They included articles in related journals, body of knowledge documents, and books. As underlying focus, the ideas developed in our research programme. At a generic knowledge level, theories related to general systems (Klir, 1969), (Klir, 1985), (Klir, 1991), (Klir and Elias, 2003), (Morbach et al., 2008), mereology and topology (Borst et al., 1995), (Borst et al., 1997), (Borst, 1997), (Guizzardi, 2005), (Keet and Artale, 2007), (Morbach et al., 2007) for the system's description were revised. Likewise, traditional systems' engineering theories were included as a metamodel for a later refinement for autonomous systems engineering (IEEE, 1990), (OMG, 2008a), (OMG, 2009b), (IEEE, 2000), (OMG, 2008b), (OMG, 2003), (Stahl and Völter, 2006), (Sanz and Rodríguez, 2008).

  At a domain-specific knowledge level, functions and capabilities to be part of an autonomous systems were considered, as described in (Rumbaugh et al., 2004), (Hernández and Hernando, 2008), (de la Mata, 2009), (UPM-ICEA-Team, 2006d), (UPM-ICEA-Team, 2006a), (López, 2007), (Sanz and Rodríguez, 2008), (Hernández et al., 2008), (Sanz et al., 2007a), (ASLab Team, 2006), (UPM-ICEA-Team, 2006c), (UPM-ICEA-Team, 2006b). Moreover, we analysed specific ontological elements for the autonomous system's engineering described in (Sanz et al., 2005), (Sanz et al., 2007b), (Rumbaugh et al., 2004), (Morbach et al., 2007), (van Lamsweerde, 2008), (Friedenthal et al., 2008), (Sanz et al., 1999), (Douglass, 2004), (Segarra, 2005), (Sanz and Rodríguez, 2008), (Estefan, 2008).

- *Methodology*: from the available methodologies and methods for ontology engineering, METHONTOLOGY (Fernández-López et al., 1997) was chosen as a starting point since:

  1. The stages for the development process are well and clearly defined in an ontology life cycle.
  2. It comprises different and further tasks to be considered, such as the ontology maintenance.
  3. The conceptualisation activity is decomposed in different detailed tasks, with a proposed order.
  4. Intermediate representations, such as tables and graphs, can be easily understood both by domain experts and ontology developers.
  5. It allows for flexibility from different viewpoints: the process (tasks can be revisited if new concepts are added), the representation (tables can be modified as needed) and evolving prototypes (a new prototype is obtained when adding, changing or removing terms).

  METHONTOLOGY proposes both a ontology development process, as well as an ontology life cycle closely intertwined. The development process refers to which activities are performed when building the ontology. The ontology life cycle identifies when these activities should be carried out, by a set of stages that define which activities to perform in each stage and how they are related. Both the development process and the life cycle activities were followed for the development of OASys. Some additional guidelines described in (Noy and McGuinness, 2001), (Mizoguchi, 2004) were also considered.

- *Formalisation*: a software engineering general- and specific-purpose language, such as UML (OMG, 2009a), (OMG, 2009b), was chosen to specify the ontology. We realised the limitations of UML for ontology development (Baclawski et al., 2002), (Gómez Pérez et al., 2004b), (Gasevic et al., 2006b). Our decision was based on the fact that our review of ontologies for autonomous systems and software engineering showed the wide use of UML to specify such ontologies (Gasevic et al., 2006a), (Tamma et al., 2005), (Ruiz and Hilera, 2006). Moreover, UML was suitable for the model-driven engineering process considered in our research programme (Sanz et al., 2009). Additionally, the Ontology Definition Metamodel (ODM) (OMG, 2009c) opened the possibility of a later formalisation of OASys using traditional ontological languages such as OWL and RDF, by using the metamodels, mapping and profiles defined in it.

4 OASys CONCEPTUALISATION

Even with the guidance of a methodology, some design decisions and trade-offs have to be considered whilst developing an ontology. Ours was not exception, and this section summarises our design decisions.

- **The Modular Structure:** the requirement to consider two different ontologies within OASys has already been justified and explained in former sections. It remains to explain the election of a modular structure for each one of them. For the ontology containing the elements for autonomous system's description, the ASys Ontology, it was clear the need to consider two different levels in the knowledge content. One, to provide concepts for a generic system, without paying attention to autonomous properties (the System Subontology). Two, to gather ontological elements specific for the kind of autonomous systems we are developing (the ASys Subontology). The first one was conceptualised to include elements that we will use to define any kind of system, i.e., even if our research changes to a different kind of systems, we will still be able to re-use this part of the ontology. The second one was conceptualised bearing in mind the necessity to provide a common vocabulary of our research view on the topic of autonomous systems.

  When it comes to the use of packages, the underlying idea was to organise the ontological constructs in a way easy to change and to update, especially for the domain-specific knowledge that would evolve as our research would do. The packages contained in a particular Subontology had to be closely related to the intended use of it.

  Hence, the System Subontology consisted of those packages related to any system's definition, in terms of their structure, behaviour and function:

  – General Systems package: to gather concepts to characterise any kind of system's structure, function and behaviour, based on General Systems Theory.

  – Mereology package: to provide general concepts for whole-part relationships, based on Mereological and Meronymic theories.

- Topology package: to collect general concepts for topological connections, based on Topological theories.

For the conceptualisation of the ASys Subontology, the inner organisation in packages follows the capabilities we consider key elements of autonomy:

- Perception package: to conceptualise the perception process.
- Knowledge package: to specify the different kinds of knowledge the autonomous system will use.
- Thought package: to describe the reconfiguration and adaption of goals in the autonomous systems.
- Action package: to characterise the way the decisions are finally carried out as actions by different actors.
- Device package: to define the aspects of the devices being part of the autonomous system.

A similar process was followed to establish the modular structure of the ASys Engineering Ontology. The name refers to the fact that this ontology provides the conceptualisation of the engineering process of an autonomous system, ASys for short.

Once again, two levels of knowledge were considered. The higher level concepts can be re-used for different engineering processes (the System Engineering Subontology), whereas the lower level ones specifically addressed the engineering process under development as part of our research on autonomous systems (the ASys Engineering Subontology). The inner structure in packages of the former subontologies was decided based on the extendibility requirement, as the ontology would be used in different applications.

The System Engineering Subontology contained those packages necessary to address the engineering process of any system:

- Requirement package: to define the stakeholders needs and requirements.
- Perspective package: to specify the stakeholders concerns.
- Engineering Process package: to describe the engineering process itself in terms of phases, tasks and products obtained.
- Model-driven package: to include model-driven theories as the overall approach.

At a specific level knowledge, we considered similar packages to specialise the former ones for our research approach, except for an ASys model-driven package, which has not yet been conceptualised until further development of our model-based control approach:

- ASys Requirement package: to specialise the stakeholders needs and requirements for an autonomous system.
- ASys Perspective package: to describe an autonomous system from different perspectives or views.
- ASys Engineering Process package: to describe the engineering process of an autonomous system.

- **The Packages' Contents:** to define the ontological elements to be considered in each one of the subontologies, and the inner packages, we followed a combination of top-down and a bottom-up approaches. The top-down approach allowed starting the ontology development with an intuitive analysis of the basic concepts and specifying them in detail afterwards. This approach was used to define the different packages to be contained in a particular subontology as described before, as well as a first overall description of the contents to be included in each one. For example, for the Thought package that conceptualises the goal-oriented process in the autonomous system, we considered at a first stage the necessity to include goal-oriented terms such as goal, subgoal, goal structure, etc.

  Next, we followed a bottom-up approach to elicit the concepts finally contained in each package, by analysing the terms actually used in a given field of knowledge and trying to interpret them and their structural relations. Continuing with the Thought package as example, we analysed goal-oriented theories and terminology on this field as described in (Yu, 1997), (van Lamsweerde, 2003), (University of Toronto, 2004). Additional techniques described in (Douglass, 2004) were used to identify the objects domain, such as underlying the nouns in the analysed texts, identifying causal objects (sources of actions or events), identifying real-world entities, physical devices, key concepts, or control elements.
- **The Concepts' Integration:** this process posed a twofold approach depending on the sources considered as input for a package. Some packages were based upon a concrete theory that provided the ontological elements. The description given by such theory was well articulated, however not being expressed from an ontological viewpoint. Key concepts were identified following the

bottom-up approach, establishing the fundamental concepts and relationships considering minimal ontological commitments, i.e., only those relevant concepts for our research and the domain of autonomous systems were initially considered.

Such is the case of the General Systems package whose main source is the General Systems Theory described in (Klir, 1969), and (Klir and Elias, 2003). The theory provides a wide range of concepts to define different features in any kind of system, however our interest lied on the necessity to use it as the basis to describe the structural, and behavioural generic knowledge, hence only those concepts related to structure and behaviour, and their relationships were considered in this package. For example, goal-oriented concepts further described in (Klir, 1991) were not included as part of this package.

For other packages, the sources and inputs for their knowledge were covered by different glossaries and theories. Hence, it was necessary among other activities to assess the granularity of the terms, the existence of synonyms, and the suitability of the concepts for our research. This assessment process was especially relevant for the domain-specific packages, where not only our research ideas but also existing sources with a similar approach had to be considered.

For example, the ASys Engineering Process package contents were obtained by analysing, mapping and manually merging the concepts described in (Sanz et al., 1999), (Segarra, 2005), combined with a review of existing model–based engineering methodologies in (Estefan, 2008).

- **The Ontologies Intradependencies:** the original design idea was to develop self-contained subontologies, i.e., grouping concepts without depending on the ontological elements of any other subontology or package. However, the layered structure into generic and domain-specific knowledge, where the latter specialises the former, showed the impossibility of such approach. Hence, intradependencies exist among the subontologies in the ASys Ontology, and in the ASys Engineering Ontology. This fact made necessary to assure the conceptualisation of the generic concept prior to the conceptualisation of specialised ones.

  For example, the taxonomy of different Quantities (variables) defined in the ASys Subontology as part of the Knowledge package, made necessary to conceptualise first the generic concept of Quantity in the System Subontology. In a similar way, the ontological constructs conceptualised in the ASys Engineering Subontology within the ASys Engineering Process package to define the different activities in the autonomous system's engineering process, depend on the former definition of the generic concepts of Phase, Task, Workproduct, etc., defined in the respective package of the System Engineering Subontology.

- **The Ontologies Interdepencies:** a second kind of dependency between the ontologies had to be considered, not so much as part of the conceptualisation of the ontologies content but to accomplish the intended use of the ontologies. The ASys Ontology conceptualises the elements to describe an autonomous system. The ASys Engineering Ontology does similarly with the terms of the autonomous system's engineering process. This process was conceptualised as different phases, tasks, and workproducts in the form of conceptual models to describe the stakeholders' needs, the autonomous system's structure, behaviour and function. These conceptual models use the ontological constructs of the ASys Ontology, thus their conceptualisation in terms of definition, attributes, relationships and axioms had to be prior made.

  For example, the ASys Engineering Ontology contains in the ASys Engineering Process package the definition of the concepts of Structural Model to model the autonomous system from a Structure viewpoint, hence the concept of Structure has previously been defined in the ASys Ontology to understand the meaning of this second concept. Likewise, the concept of Operation Model relies on the definition of the Operation concept, made in the Action package of the ASys Ontology.

  These interdependencies were addressed and described in an ontology-based methodology, which describes and guides the conceptual modelling of an autonomous system based on the ontological constructs provided by the ASys Engineering Ontology, which in turn uses the elements in the ASys Ontology.

5 OASys FORMALISATION

Considering the requirements, their fulfilment and the design decisions described in former sections, the final ontology for autonomous systems (OASys) was formalised as two main ontologies: the ASys Ontology for the ontological elements related to the system's description, and the ASys Engineering Ontology to provide system's engineering ontological constructs. Each one of them was conceptualised and formalised as a standalone ontology, using the chosen methodology.

Hence, OASys is in fact two ontologies grouped under the same name. However, they were conceived to be used in conjunction, with the ASys Engineering Ontology contents constructing and guiding the use of the ASys Ontology contents during the process of autonomous system's conceptual modelling.

1. **ASys Ontology:** as part of it, two subontologies were developed to cover from generic knowledge to domain–specific one regarding autonomous system's description (Figure 1). The *System Subontology* contains the generic knowledge on systems, organised into the General Systems, Mereology, and Topology packages. The *ASys Subontology* specialises and refines the previous concepts, adding autonomous systems specific ones, consisting of the Perception, Knowledge, Thought, Action, and Device packages.

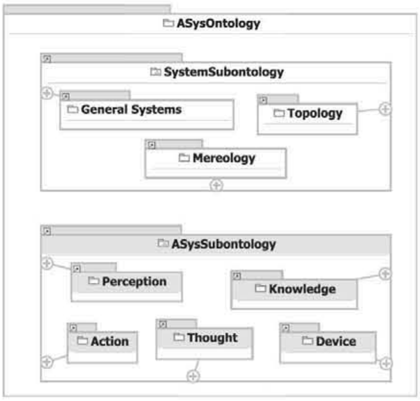

Figure 1: The ASys Ontology.

2. **ASys Engineering Ontology:** two different subontologies were developed as part of this ontology to conceptualise the engineering process of autonomous systems, from a more abstract to domain–specific knowledge (Figure 2). The *System Engineering Subontology* gathers ontological elements for any system engineering process as general as possible based on system's engineering and software engineering methodologies, organised into the Requirement, System Perspective, Engineering Process and Model–driven packages. The *ASys Engineering Subontology* contains the specialisation and additional elements to describe an autonomous system's generic engineering process, consisting of the ASys Requirement, ASys Perspective, and ASys Engineering Process packages.

To assess the suitability and shortcomings of the ontology and the related methodology, two testbeds have been considered to obtain the conceptual models (Bermejo-Alonso et al., 2010c).

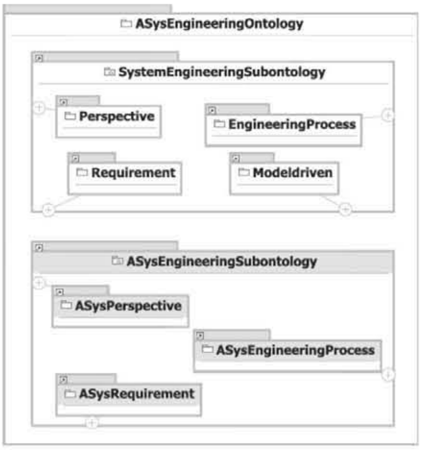

Figure 2: The ASys Engineering Ontology.

The first one, the Robot Control Testbed (RCT), is a collection of mobile robot systems, with a wide range of implementations and capabilities (from conventional SLAM based mobile robots to virtual ones inspired in rat brain neuroscience).

A second testbed, the Process Control Testbed (PCT), involves the development of a robust control architecture for a chemical reaction system (with multiple steady states), providing the system with cognitive capabilities to carry out complex tasks such as fault diagnosis, alarm management, and control system reconfiguration from a single theoretical standpoint.

6 CONCLUSIONS

Our research focused on the engineering and development of a modular ontology, as a set of smaller and interrelated ontologies, to be used as a conceptual framework and software support for the domain of autonomous systems. This paper describes the engineering process of such ontology, paying special attention to the requirements, the features, and the structure of the ontology. Once these elements were defined, the conceptualisation and formalisation stages of the ontology took place. As a result, the ontology for autonomous systems (OASys) was obtained.

This ontology is the initial step in a broader research aim to develop autonomous systems where such systems will use their own design knowledge during their operation. This knowledge will be represented in the form of conceptual models based on the ontological elements contained in the ontology described in this paper.

Hence, the ontological terms will be initially used to describe the autonomous system's features and functionalities (by means of the ASys Ontology) and the engineering process (by means of the ASys Engineering Ontology) (Bermejo-Alonso et al., 2010b).

The ASys Ontology will allow us to describe different kinds of autonomous systems, both at a general and at a detailed knowledge level to consider the different elements and processes we consider of importance in our autonomous systems.

The ASys Engineering Ontology foresees the necessity to conceptualise our new approach for engineering autonomous systems at a more detailed knowledge level, however considering generic engineering elements to describe the process at a more abstract level.

By using OASys, engineers developing autonomous' systems will obtain the conceptual models of different autonomous systems, addressing the problem of modelling in a modular and unified view the complex systemic structures that many autonomous systems exhibit.

Additionally, OASys has been complemented with the development of the OASys-based methodology to exemplify the use of OASys in a generic autonomous system engineering process (Bermejo-Alonso et al., 2010a).

The methodology will guide autonomous system's engineers on how the conceptual models of an autonomous system can be obtained and used to describe and to support the autonomous system's engineering process. Later on, these conceptual models will be used by the autonomous system itself as knowledge to perform their operation, following a model-based control paradigm.

The key aspect of this research is that to develop OASys we considered a wide range of autonomous systems, covering from robot-based applications to continuous processes or biological systems. Former attempts of ontologies for this domain focused on a particular kind of application or a specific type of autonomous systems.

Moreover, the development of OASys has addressed not only its use for the autonomous systems' description but also for their engineering process. The combination of these two aspects was not considered in previous research efforts regarding these domains.

To consider the autonomous systems' domain from a global viewpoint made it necessary to take into account a great number of sources and elements to make the ontology as comprehensive as possible. OASys has allowed us to conceptualise the domain of autonomous systems in a way general and reusable enough to address any kind of autonomous system. The ontology has conceptualised domain knowledge both at a general level and at a more specific level, without being application-oriented.

This approach has allowed us to model the testbeds at the level of detail required for their software development. However, the particular features of the testbeds have hinted a possible necessity to complement our ontology with subdomain or application specific knowledge. This will lead to additional analysis, mapping and integration aspects to be addressed as part of further research.

The ontology structure was chosen considering the different requirements, to cater for different levels in the contents as well as different domains in use. The modelling of the testbeds using OASys showed the suitability of this multilevel modular approach, although pinpointing the complexity of considering in detail the dependencies among the packages and the two inner ontologies. The packaged structure allowed us to add new packages as our research evolved.

The relationship and interaction between the ASys Ontology and the ASys Engineering Ontology has been addressed in the OASys-based methodology, where it is described the use of the ASys Ontology elements as part of an engineering process defined using the ASys Engineering Ontology.

Further research will also address the evaluation process of the adopted ontology, considering different available methods and techniques.

Our research also needs to explore in more detail the aspects of modularity, as well as reviewing the evolution of OASys, as a set of interconnected ontologies, into a network ontology as described in (Suarez-Figueroa et al., 2009).

REFERENCES

Abran, A., Cuadrado, J., García-Barriocanala, E., Mendes, O., Sánchez-Alonso, S., and Sicilia, M. (2006). Engineering the ontology for the SWEBOK: Issues and techniques. In Calero, C., Ruiz, F., and Piattini, M., editors, *Ontologies for Software Engineering and Software Technology*, chapter 3, pages 103–121. Springer-Verlag Berlin Heidelberg.

ASLab Team (2006). Core mental terminology: from an autonomous system perspective. Technical Report R-2006-XXX, Autonomous Systems Laboratory (ASLab).

Baclawski, K., Kokar, M., Kogut, P., Hart, L., Smith, J., Letkowski, J., and Emery, P. (2002). Extending the unified modeling language for ontology development. *Software Systems Modeling*, (1):142–156.

Barbera, T., Albus, J., Messina, E., Schlenoff, C., and Horst, J. (2004). How task analysis can be used to derive and organize the knowledge for the control of autonomous vehicles. *Robotics and Autonomous Systems*, 49:67–78.

Bermejo-Alonso, J., Sanz, R., Rodríguez, M., and Hernández, C. (2010a). An ontological framework for autonomous systems modelling. *International Journal on Advances in Intelligent Systems*, 3(3 and 4):211–225.

Bermejo-Alonso, J., Sanz, R., Rodríguez, M., and Hernández, C. (2010b). An ontology–based approach for autonomous systems' description and engineering: the OASys Framework. In Setchi, R., Jordanov, I., Howlett, R. J., and Jain, L. C., editors, *14th International Conference on Knowledge–Based and Intelligent Information and Engineering Systems (KES 2010)*, volume 6276 of *LNAI*, pages 522–531, Cardiff, Wales, U.K. Springer, Heidelberg.

Bermejo-Alonso, J., Sanz, R., Rodríguez, M., and Hernández, C. (2010c). Ontology–based engineering of autonomous systems. In Bauer, M., Mauri, J. L., and Dini, O., editors, *Proceedings of the The Sixth International Conference on Autonomic and Autonomous Systems (ICAS 2010)*, pages 47–51, Cancun, Mexico. IEEE Computer Society.

Borst, P., Akkermans, H., and Top, J. (1995). The PhysSys ontology for physical systems. In *Ninth International Workshop Ninth International Workshop on Qualitative Reasoning*, pages 11–21. Department of Social Science Informatics (S .W.I .) University of Amsterdam, Amsterdam, the Netherlands.

Borst, P., Akkermans, H., and Top, J. (1997). Engineering ontologies. *International Journal of Human-Computer Studies*, 46:365–406.

Borst, W. N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* PhD thesis, Centre for Telematics and Information Technology, University of Tweenty. Enschede. The Netherlands.

Corcho, O., Fernández-López, M., and Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & Knowledge Engineering*, 46:41–64.

de la Mata, J. L. (2009). CSTR overall specification: The main PCT testbed. Technical Report R-2009-001, Autonomous Systems Laboratory (ASLab).

Douglass, B. P. (2004). *Real Time UML: advances in the UML for real–time systems.* The Addison–Wesley object technological. Addison-Wesley, 3rd edition.

Eberhart, A. (2003). *Ontology-Based Infrastructure for Intelligent Applications.* Phd thesis, University of Saarbrücken.

Estefan, J. A. (2008). Survey of model–based systems engineering (MBSE) methodologies. Technical Report INCOSE-TD-2007-003-01 (Rev. B), Model-Based Systems Engineering (MBSE) Initiative, International Council on Systems Engineering (INCOSE).

Falbo, R., Ruy, F., and Moro, R. (2005). Using ontologies to add semantics to a software engineering environment. In *Proceedings of 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'2005)*, pages 151–156, Taipei, China.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). METHONTOLOGY: from ontological art towards ontological engineering. In Farquhar, A., Grüninger, M., Gómez-Pérez, A., Uschold, M., and van der Vet, P., editors, *AAAI'97 Spring Symposium on Ontological Engineering*, pages 33–40, Stanford University, CA, U.S.A.

Friedenthal, S., Moore, A., and Steiner, R. (2008). *A practical guide to SysML: The Systems Modeling Language.* Morgan Kaufmann and OMG Press.

Gasevic, D., Djuric, D., and Devedzic, V. (2006a). Mappings of mda-based languages and ontologies. In *Model Driven Architecture and Ontology Development*, chapter 10. Springer-Verlag Berlin Heidelberg.

Gasevic, D., Djuric, D., and Devedzic, V. (2006b). *Model Driven Architecture and Ontology Development.* Springer–Verlag Berlin Heidelberg.

Gómez Pérez, A., Fernández López, M., and Corcho, M. (2004a). Methodologies and methods for building ontologies. In *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, Advanced Information and Knowledge Processing, chapter 3, pages 107–197. Springer.

Gómez Pérez, A., Fernández López, M., and Corcho, M. (2004b). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web.* Advanced Information and Knowledge Processing. Springer.

Gruber, T. (1993). Toward principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R., editors, *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*, Padova, Italy. Kluwer Academic Publishers.

Grüninger, M. and Fox, M. (1995). Methodology for the design and evaluation of ontologies. In Skuce, D., editor, *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*, pages 6.1–6.10, Montreal, Canada.

Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models.* Phd thesis, University of Twente, The Netherlands.

Hallam, J. and Bruynickx, H. (2006). An ontology of robotics science. In Christensen, H. I., editor, *Proceedings of the European Robotics Symposium 2006 (STAR 22)*, pages 1–14. Springer–Verlag Berlin Heidelberg.

Hernández, C. and Hernando, A. (2008). RCT overall specification: Higgs platform. Technical Report R-2008-XXX, Autonomous Systems Laboratory (ASLab).

Hernández, C., Sanz, R., and López, I. (2008). Consciousness in cognitive architectures: a principled analysis of RCS, Soar and ACT–R. Technical Report R–2008–004, Autonomous Systems Laboratory (ASLab).

Hesse, W. (2005). Ontologies in the software engineering process. In Lenz, R., editor, *Proceedings of Tagungsband Workshop on Enterprise Application Integration (EAI2005)*, Berlin, Germany. GITO–Verlag.

Hruby, P. (2005). Ontology–based domain–driven design. In *Proceedings of Object-Oriented Programming, Systems, Languages And Applications (OOPSLA'05)*, San Diego, California, U.S.A.

IEEE (1990). *IEEE Standard Glossary of Software Engineering Terminology.* IEEE Computer Society, New York, IEEE std 610.12 1990 edition.

IEEE (2000). *IEEE Recommended Practice for Architectural Description for Software- Intensive Systems.* Institute for Electrical and Electronics Engineering, New York, IEEE std 1471-2000 edition.

Keet, C. and Artale, A. (2007). Representing and reasoning over a taxonomy of part–whole relations. *Applied Ontology*, 0(1):1–17.

Klir, G. J. (1969). *Approach to General Systems Theory.* Van Norstrand Reinhold, New York.

Klir, G. J. (1991). *Facets of Systems Science.* Plenum Press.

Klir, G. J. and Elias, D. (2003). *Architecture of Systems Problem Solving*, volume 21 of *IFSR International Series on Systems Science and Engineering.* Kluwer Academic Publishers.

Klir, G. J. K. (1985). The emergence of two–dimensional science in the information society. *Systems Research*, 2(1):33–41.

Lehtihet, E., Strassner, J., Agoulmine, N., and Foghlu, M. O. (2006). Ontology-based knowledge representation for self-governing systems. In State, R., editor, *Proceedings of the 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2006)*, volume LNCS 4269. IFIP International Federation for Information Processing.

López, I. (2007). *A Foundation for Perception in Autonomous Systems.* Phd thesis, Departamento de Automática, Ingeniería Electrónica e Informática Industrial, Universidad Politécnica de Madrid.

Malucelli, A., Palzer, D., and Oliveira, E. (2005). Combining ontologies and agents to help in solving the heterogeneous problem in e-commerce negotiations. In *International Workshop on Data Engineering Issues in E-Commerce (DEEC 2005), IEEE Computer Society*, pages 26–35, Tokyo, Japan.

Mizoguchi, R. (2004). Tutorial on ontological engineering - part 2: ontology development, tools and languages. *New Generation Computing*, 22(1):61–96.

Morbach, J., Bayer, B., Wiesner, A., Yang, A., and Marquardt, W. (2008). OntoCAPE 2.0: the upper level. Technical Report LPT–2008–25, RWTH Aachen University.

Morbach, J., Wiesner, A., and Marquardt, W. (2007). A meta model for the design of domain ontologies. Technical Report LPT-2008-24, RWTH Aachen University.

Noy, N. and McGuinness, D. (2001). Ontology development 101: A guide to creating your first ontology. Technical Report KSL–01–05, Stanford Knowledge Systems Laboratory.

OMG (2003). *MDA Guide Version 1.0.1.* Object Management Group.

OMG (2008a). *OMG SysML Specification.* Object Management Group, v 1.1 edition.

OMG (2008b). Software and systems process engineering meta–model specification version 2.0. OMG Formal Specification 2008–04–01, Object Management Group, Inc.

OMG (2009a). *OMG Unified Modeling Language (OMG UML) Infrastructure Version 2.2.*

OMG (2009b). *OMG Unified Modeling Language (OMG UML) Superstructure Version 2.2.*

OMG (2009c). *Ontology Definition Metamodel Version 1.0.* Object Management Group.

Provine, R., Uschold, M., and Smith, S. (2004). Observations on the use of ontologies for autonomous vehicle navigation planning. In *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, Stanford, California.

Ruiz, F. and Hilera, J. (2006). *Ontologies for Software Engineering and Software Technology.* Springer-Verlag Berlin Heidelberg.

Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *The Unified Modeling Language Reference Manual.* Object Technology. Addison-Wesley, second edition.

Sanz, R., Hernández, C., Gomez, J., Bermejo-Alonso, J., Rodríguez, M., Hernando, A., and Sanchez, G. (2009). Systems, models and self–awareness: towards architectural models of consciousness. *International Journal of Machine Consciousness*, 1(2):255–279.

Sanz, R., Hernández, C., and Rodríguez, M. (2007a). ASys models: Model–driven engineering in ASys. Technical Report R–2007–016, Autonomous Systems Laboratory (ASLab).

Sanz, R., López, I., and Bermejo, J. (2007b). *Artificial Consciousness*, chapter A rationale and vision for machine consciousness in complex controllers, pages 141–155. Imprint Academic, Exeter, UK.

Sanz, R., López, I., Bermejo, J., Chinchilla, R., and Conde, R. (2005). Self-X: The control within. In *Proceedings of the 16th IFAC World Congress*, Praga, Czech Republic. IFAC.

Sanz, R., Matia, F., and Puente, E. A. (1999). *Microprocessor–based and intelligent systems engineering*, chapter The ICa approach to intelligent autonomous systems. Kluwer Academic Publishers.

Sanz, R. and Rodríguez, M. (2008). The ASys vision: Engineering any-x autonomous system. Technical Report R-2007-001, Autonomous Systems Laboratory (ASLab).

Schlenoff, C. and Messina, E. (2005). A robot ontology for urban search and rescue. In *Proceedings of the 2005 ACM workshop on Research in knowledge representation for autonomous systems*, pages 27–34, Budapest, Hungary. ACM Press.

Scrapper, C. and Balakirsky, S. (2004). Knowledge representation for on–road driving. In *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, Stanford, California.

Segarra, M. J. (2005). *CORBA control systems.* Phd thesis, Universidad Politécnica de Madrid.

Stahl, T. and Völter, M. (2006). *Model-Driven Software Development: technology, engineering, management.* John Wiley and Sons, Ltd.

Stojanovic, L., Abecker, A., Stojanovic, N., and Studer, R. (2004a). Ontology–based correlation engines. In Computer, I., editor, *Proceedings of the International*

Conference on Autonomic Computing (ICAC'04), pages 304–305.

Stojanovic, L., Schneider, J., Maedche, A., Libischer, S., Studer, R., Lumpp, T., Abecker, A., Breiter, G., and Dinger, J. (2004b). The role of ontologies in autonomic computing systems. *IBM Systems Journal*, 43(3):598 – 616.

Suarez-Figueroa, M. C., Blomqvist, E., D'Aquin, M., M.Espinoza, Gómez-Pérez, A., Lewen, H., Mozetic, I., Palma, R., Poveda, M., Sini, M., Villazón-Terrazas, B., Zablith, F., and Dzbor, M. (2009). Revision and extension of the NeOn methodology for building contextualized ontology networks. Technical Report D 5.4.2, Neon Project.

Tamma, V., Cranefield, S., Finin, T., and Willmott, S., editors (2005). *Ontologies for Agents: Theory and Experiences*. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkhäuser.

Tziallas, G. and Theodoulidis, B. (2003). Building autonomic computing systems based on ontological component models and a controller synthesis algorithm. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, pages 674–680, Prague, Czech Republic.

University of Toronto (2004). GRL ontology.

UPM-ICEA-Team (2006a). Case studies of perception and system analysis. Technical Report ASLab-ICEA-R-2006-015, 1.0 Final, Autonomous Systems Laboratory (ASLab).

UPM-ICEA-Team (2006b). ICEA glossary: integration, cognition, emotion, autonomy. Technical Report ASLab-ICEA-R-2006-014, Autonomous Systems Laboratory (ASLab).

UPM-ICEA-Team (2006c). A vision of general autonomous systems. Technical Report ASLab-ICEA-R-2006-018, 1.0 Final, Autonomous Systems Laboratory (ASLab).

UPM-ICEA-Team (2006d). A vision of perception in autonomous systems. Technical Report ASLab-ICEA-R-2006-017, Autonomous Systems Laboratory (ASLab).

Uschold, M. and King, M. (1995). Towards a methodology for builiding ontologies. In Skuce, D., editor, *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*, pages 6.1–6.10, Montreal, Canada.

Uschold, M., Provine, R., Smith, S., Schlenoff, C., and Balikirsky, S. (2003). Ontologies for world modeling in autonomous vehicles. In *18Th International Joint Conference on Artificial Intelligence, IJCAI'03*.

van Lamsweerde, A. (2003). From system goals to software architecture. In Bernardo, M. and Inverandi, P., editors, *Formal Methods for Software Architecture*, volume LNCS 2804. Springer Verlag.

van Lamsweerde, A. (2008). Requirements engineering: From craft to discipline. In *Proceedings of FSE'2008: 16th ACM Sigsoft International Symposium on the Foundations of Software Engineering*, Atlanta, U.S.A.

Wongthongtham, P., Chang, E., and Dillon, T. (2005). Towards ontology-based software engineering for multisite software development. In *Proceedings of 3rd IEEE International Conference on Industrial Informatics (INDIN)*, pages 362–365, Perth, Australia.

Yu, E. (1997). Towards modelling and reasoning support for early–phase requirements engineering. In *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, pages 226–235, Washington, D.C., USA.