Ontology Engineering for the Autonomous Systems Domain

Julita Bermejo–Alonso, Ricardo Sanz, Manuel Rodríguez, and Carlos Hernández

Autonomous Systems Laboratory (ASLab), Universidad Politécnica de Madrid, Spain
jbermejo@etsii.upm.es,
{ricardo.sanz,manuel.rodriguezh,carlos.hernadez}@upm.es
http://www.aslab.org

Abstract. Ontologies provide a common conceptualisation that can be shared by all stakeholders in an engineering development process. They provide a good means to analyse the domain, allowing to separate descriptive from problem–solving knowledge. Our research programme on autonomous systems considered an ontology as the adequate mechanism to conceptualise the autonomous systems domain, and the software engineering techniques applied to such systems. This paper describes the ontological engineering process of such an ontology: OASys (Ontology for Autonomous Systems). Its development considered different stages: the specification of the requirements to be fulfilled by the ontology; the extraction of the actual features needed to implement the desired requirements; the conceptualisation phase with the design decisions to integrate the different domains, theories and techniques addressed by the ontological elements; and finally, the implementation of the ontology, which integrates both ontology engineering and software engineering approaches by using UML as the implementation language.

Keywords: Ontological engineering, Knowledge-based engineering, Autonomous systems.

1 Introduction

Knowledge Engineering research has addressed the use and the development of ontologies as a means to improve knowledge processes. An ontology as a conceptualisation of a specification [25], provides a solid basis to build knowledge bases for a greater functionality 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

A. Fred et al. (Eds.): IC3K 2011, CCIS 348, pp. 263–277, 2013.

an ontology [23]. This paper describes how we carried out the ontological engineering of an ontology for autonomous systems. We have developed this 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 one on autonomous systems, providing a common vocabulary for all the stakeholders. The underlying idea is that the ontology should express the concepts and consider the constraints or relationships in an explicit way under some ontological commitments, as the ontology will 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, with the ontological concepts being used at run-time by the autonomous system. 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 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 to refine it.

2 Related Work

Our ontology for autonomous systems (OASys) addressed two different but interrelated domains. Firstly, the domain of autonomous systems. Secondly, the domain of software engineering.

Related to the domain of autonomous systems, ontologies have addressed different kinds of autonomous systems: mobile robots, agent-based applications, and autonomic systems. 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 [76], on their use for a particular mobile robot or application [4], [63], [62], [28], and on the benefits achieved [67], [53]. For agent-based systems, the research on ontologies emphasises the necessity to share and to exchange knowledge among the agents in the system, and the problems of interoperatibility [42], [62]. In autonomic systems, ontologies support information exchange and integration [40], as part of the autonomous system [69], and as an explicit representation of data semantic and rules [66]. 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 [31], [54], [1], [32], [18]. Additionally, ontologies have been used as software elements within the system's architecture to support the software process [16], [79]. 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 systems and for software engineering [7], 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 [75], [46], [13], [26], 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 fulfil the requirements of the ontology to be used for the description and engineering of autonomous systems.

3 OASys Specification

A key aspect whilst developing an ontology is to state its purpose, which drives the development and its 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 [25]. 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.

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 both to describe and to engineer the autonomous system.
- *Type of ontology:* it would be a domain ontology to describe the autonomous system 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 constrained to autonomous systems related aspects.
- *Design criteria:* to assure its coherence and quality, the development would respect design criteria such as: 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 serve as an input source for the ontological elements. 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 source, since they provide their terminology in a domain they are familiar with.
– *Methodology:* the election of the methodology to follow during the ontology building is also an important factor. There is a wide range of methodologies to support and guide this process, as reviewed in [22], [48]. It would be necessary to assess them, to be reused in the ontology development.
– *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 should be made to select the most convenient.

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, the ontology has adopted a layered structure to address both generic and domain-specific knowledge. The upper layer contains the more abstract level knowledge. A lower layer gathers the ontological elements to charaterise an autonomous' system structure, function and behaviour. 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 and the ASys Engineering Ontology.
– *Design Criteria:* the design criteria were followed throughout the development of the ontology. To address the *clarity* criteria, existing ontologies and glossaries were reviewed to extract the ontological elements. Those concepts would be later discussed with the group members to commit to the desired meaning for our research, and defined in natural language. To cater for extensions in the future for the *extendibility* criteria, subontologies and packages organised the concepts. Subontologies group ontological elements at the different abstraction levels. Packages classify the concepts within a subontology according to a concrete aspect. These organising elements allow the extension or modification of the ontology without major

changes to its structure and composition. To consider the *minimal encoding bias*, intermediate tabular representations and graphs were used to define the different ontological constructs. *Minimal ontological commitment* was achieved considering only the fundamental concepts as agreed by the ontology users both at a generic knowledge, and at a domain–specific level.

– *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. The different sources were analysed to extract the ontological elements, checking for commonalities, mismatches and level of granularity. Experts were also questioned for their domain knowledge, as an additional input for ontological concepts.
– *Methodology:* from the available methodologies and methods, METHONTOLOGY [19] was chosen as a starting point for several reasons. First, the stages for the development process are well and clearly defined in an ontology life cycle. It also comprises different and further tasks to be considered, such as the ontology maintenance. Moreover, the conceptualisation activity is decomposed in different detailed tasks, with a proposed order. As a key element, intermediate representations, such as tables and graphs, can be easily understood both by domain experts and ontology developers. Finally, the methodology allows for flexibility in the process, the representation and the existence of evolving prototypes. Some additional guidelines described in [46], [43] were also considered.
– *Formalisation:* a software engineering general- and specific-purpose language, such as UML [51], was chosen to specify the ontology. We realised the limitations of UML for ontology development [3], [23]. Our decision was based on our review of ontologies for autonomous systems and software engineering where UML has been widely used [21], [68], [54]. Additionally, the Ontology Definition Metamodel (ODM) [52] 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

Our ontological engineering process addressed the requirements and the features of the ontology described in Section 3. The specified characteristics were actually conceptualised at this stage of the development of the ontology. Even with the guidance of a methodology, some design decisions and trade-offs had to be considered whilst developing our ontology:

– **The Modular Structure.** 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 level provided concepts for a generic system, without paying attention to autonomous properties (the System Subontology). A second level gathered ontological elements specific for the the capabilities

Table 1. ASys Ontology: Packages and Sources

SUBONTOLOGY	PACKAGE	PURPOSE	SOURCES
System	General Systems	To characterise system's structure, function and behaviour	[36], [39], [37], [38],
	Mereology	To represent whole-part relationships	[8], [10], [27], [35], [45], [44]
	Topology	To describe topological connections	[9], [45], [44]
ASys	Perception	To conceptualise the perceptive and sensing processes	[74], [71], [41]
	Knowledge	To consider the different kinds of knowledge an autonomous system uses	[61], [41], [30], [57], [2]
	Thought	To describe the reconfiguration and adaptation of goals	[73], [72], [80], [77], [70]
	Action	To specify the actions different actors will carry out	[41], [30]
	Device	To define the devices features	[55], [29], [14]

we consider key elements of autonomy in our research (the ASys Subontology, where ASys stands for autonomous system). To define the packages in each subontology, 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, their purpose and the sources are shown in Table 1.

A similar process was followed to establish the modular structure of the ASys Engineering Ontology. 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 final considered packages are shown in Table 2.

– **The Packages' Contents:** to define the ontological elements to be considered in each one of the subontologies, 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 general 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 [80],

Table 2. ASys Engineering Ontology: Packages and Sources

SUBONTOLOGY	PACKAGE	PURPOSE	SOURCES
System Engineering	Requirement	To define stakeholders need and requirements	[33], [49], [51], [61]
	Perspective	To specify stakeholders concerns	[33], [34], [20]
	Engineering Process	To describe the engineering process as phases, tasks and products obtained	[50]
	Model–driven	To include model–driven theories	[47], [65]
ASys Engineering	ASys Requirement	To specialise stakeholders requirements for an autonomous system	[59], [58], [30]
	ASys Perspective	To describe and autonomous system from different perspectives	[55], [45], [78], [20]
	ASys Engineering Process	To describe the engineering process of an autonomous system	[60], [15], [64], [61], [17]

[77], [70]. Additional techniques described in [15] 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, 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. For other packages, the knowledge was covered by several sources, being necessary 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 [60], [64], combined with a review of existing model–based engineering methodologies in [17].
– **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, this was not possible for two reasons. Firstly, the definition of some concepts in one package was based upon concepts belonging to another package. Secondly, the layered structure into generic and domain-specific knowledge, made necessary to assure the conceptualisation of the generic concept prior to the domain specialised ones.

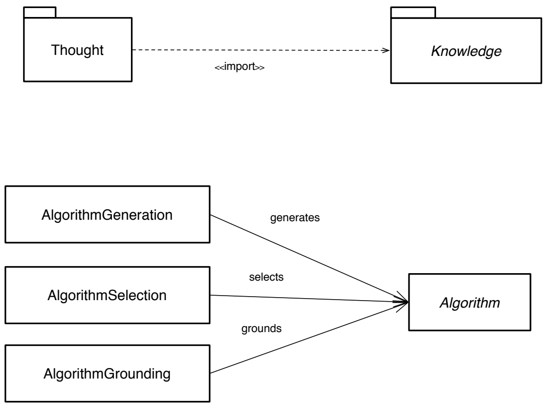

Fig. 1. Thought package usage of Knowledge package

For example, the Thought package defines the concepts of AlgorithmGeneration, AlgorithmSelection and AlgorithmGrounding as phases in the thinking process of an autonomous system. Their definition is based upon the concept Algorithm, that had to be previously define in the Knowledge package as a kind of knowledge in the autonomous system (Figure 1).

– **The ontologies interdependencies**: 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 previously made. These interdependencies were addressed and described in an ontology-based methodology [5], 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 (Figure 2).

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 (Figure 3): 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

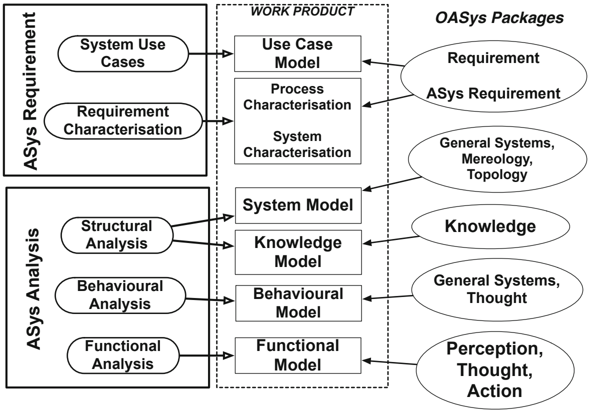

Fig. 2. OASys-based Methodology: phases, workproducts and OASys related packages

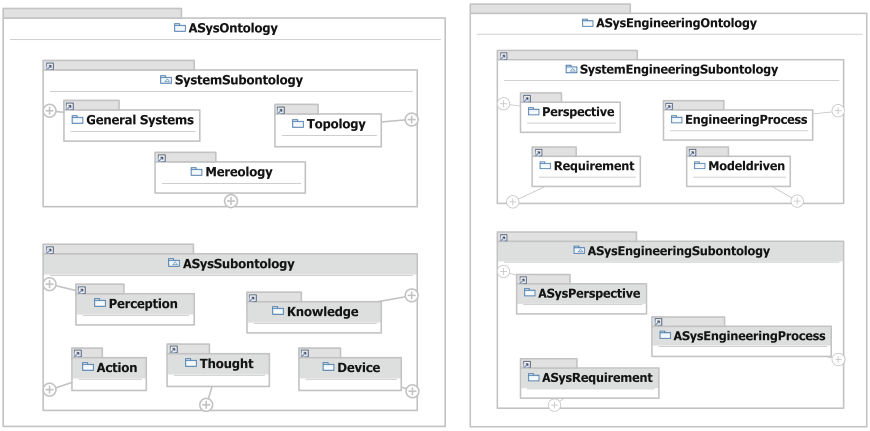

Fig. 3. OASys formalisation: ontologies, subontologies and packages

guiding the use of the ASys Ontology contents during an 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. 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.
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. 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, 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.

6 Concluding Remarks

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. The ontology is the initial step in a broader research aiming at developing 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 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). 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.

OASys has been complemented with the development of an OASys-based methodology to support for the use of OASys in a generic autonomous system engineering process [7]. The OASys–based Engineering Methodology is a generic ontology-based autonomous systems development methodology based on the OASys ontological elements. The methodology focuses on the description on how to carry out the engineering process of an autonomous system, in terms of phases, tasks, work products, having as guideline the ontological elements in the System Engineering and ASys Engineering subontologies. Being OASys–based, the methodology considers the ontological elements required in the different tasks, by specifying the OASys packages to be used. The outcome of applying the methodology is a set of conceptual models that show the structural, behavioural and functional features of the autonomous system under study. 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.

To assess the suitability and shortcomings of the ontology and the related methodology, two testbeds have been considered to obtain these conceptual models [5], [6]. 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.

The application of the framework composed of the ontology (OASys) and the methodology (OASys–based Engienering Methodology) has allowed us to evaluate and to assess the ontological contents from the users' point of view. From the different ontology evaluation techniques [24], [12], [11] we adopted an application–based evaluation that consists in using the ontology in an application and evaluating the results. The goal is to determine what has been correctly defined by the ontology, what has not been defined in the ontology, and what has been incorrectly defined. By applying the ontology, we checked the consistency, completeness, and conciseness of the ontologies, subontologies, and packages.

The first conclusion from evaluating the ontology was that it 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. For example the different data types corresponding to the different sensors in mobile robots had to be conceptualised and integrated in subpackages under the OASys Knowledge package. Secondly, the ontology structure was chosen 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. As example, the Epistemic Control Loop package has been developed to conceptualise the operation of control loops based on models according to the ECL model [56], with dependencies from the Though package and others from the ASys Ontology.

References

1. Abran, A., Cuadrado, J., García-Barriocanala, E., Mendes, O., Sánchez-Alonso, S., Sicilia, M.: Engineering the ontology for the SWEBOK: Issues and techniques. In: Calero, C., Ruiz, F., Piattini, M. (eds.) Ontologies for Software Engineering and Software Technology, pp. 103–121. Springer, Heidelberg (2006)
2. ASLab Team: Core mental terminology: from an autonomous system perspective. Technical Report R-2006-XXX, Autonomous Systems Laboratory (ASLab) (2006)
3. Baclawski, K., Kokar, M., Kogut, P., Hart, L., Smith, J., Letkowski, J., Emery, P.: Extending the unified modeling language for ontology development. Software Systems Modeling 1, 142–156 (2002)
4. Barbera, T., Albus, J., Messina, E., Schlenoff, C., Horst, J.: 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 (2004)
5. Bermejo-Alonso, J., Sanz, R., Rodríguez, M., Hernández, C.: An Ontological Framework for Autonomous Systems Modelling. International Journal on Advances in Intelligent Systems 3, 4, 211–225 (2010)
6. Bermejo-Alonso, J., Sanz, R., Rodríguez, M., Hernández, C.: An Ontology–Based Approach for Autonomous Systems' Description and Engineering: The OASys Framework. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010, Part I. LNCS, vol. 6276, pp. 522–531. Springer, Heidelberg (2010)

7. Bermejo-Alonso, J.: OASys: ontology for Autonomous Systems. PhD thesis, Universidad Politécnica de Madrid (2010)
8. Borst, P., Akkermans, H., Top, J.: The PhysSys ontology for physical systems. In: Ninth International Workshop Ninth International Workshop on Qualitative Reasoning. Department of Social Science Informatics (S.W.I.) University of Amsterdam, Amsterdam, The Netherlands, pp. 11–21 (1995)
9. Borst, P., Akkermans, H., Top, J.: Engineering ontologies. International Journal of Human-Computer Studies 46, 365–406 (1997)
10. Borst, W.N.: Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PhD thesis, Centre for Telematics and Information Technology, University of Tweenty, Enschede, The Netherlands (1997)
11. Obrst, L., Ashpole, B., Ceusters, W., Mani, I., Ray, S.R., Smith, B.: The Evaluation of Ontologies. In: Baker, C.J.O., Cheung, K.-H. (eds.) Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences, 1st edn. Springer (2006)
12. Brank, J., Grobelnik, M., Mladenic, D.: A survey of ontology evaluation techniques. In: Conference on Data Mining and Data Warehouses, SiKDD (2005)
13. Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies. where is their meeting point? Data and Knowledge Engineering 46, 41–64 (2003)
14. de la Mata, J.L.: CSTR overall specification: The main PCT testbed. Technical Report R-2009-001, Autonomous Systems Laboratory, ASLab (2009)
15. Douglass, B.P.: Real Time UML: advances in the UML for real–time systems, 3rd edn. The Addison–Wesley object technological. Addison-Wesley (2004)
16. Eberhart, A.: Ontology-Based Infrastructure for Intelligent Applications. Phd thesis, University of Saarbrücken (2003)
17. Estefan, J.A.: 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) (2008)
18. Falbo, R., Ruy, F., Moro, R.: Using ontologies to add semantics to a software engineering environment. In: 17th International Conference on Software Engineering and Knowledge Engineering (SEKE 2005), Taipei, China, pp. 151–156 (2005)
19. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: from ontological art towards ontological engineering. In: Farquhar, A., Grüninger, M., Gómez-Pérez, A., Uschold, M., van der Vet, P. (eds.) AAAI 1997 Spring Symposium on Ontological Engineering, Stanford University, CA, U.S.A, pp. 33–40 (1997)
20. Friedenthal, S., Moore, A., Steiner, R.: A practical guide to SysML: The Systems Modeling Language. Morgan Kaufmann and OMG Press (2008)
21. Gasevic, D., Djuric, D., Devedzic, V.: Model Driven Architecture and Ontology Development. Springer, Heidelberg (2006)
22. Gómez Pérez, A., Fernández López, M., Corcho, M.: 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, pp. 107–197. Springer (2004)
23. Gómez Pérez, A., Fernández López, M., Corcho, M.: Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Advanced Information and Knowledge Processing. Springer (2004)
24. Gómez Pérez, A.: Ontology Evaluation. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 251–274. Springer (2004)
25. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. In: Guarino, N., Poli, R. (eds.) International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation, Padova, Italy. Kluwer Academic Publishers (1993)

26. Grüninger, M., Fox, M.: Methodology for the design and evaluation of ontologies. In: Skuce, D. (ed.) IJCAI 1995 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, pp. 6.1–6.10 (1995)
27. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Phd thesis, University of Twente, The Netherlands (2005)
28. Hallam, J., Bruynickx, H.: An ontology of robotics science. In: Christensen, H.I. (ed.) European Robotics Symposium 2006 (STAR 22), pp. 1–14. Springer, Heidelberg (2006)
29. Hernández, C., Hernando, A.: RCT overall specification: Higgs platform. Technical Report R-2008-XXX, Autonomous Systems Laboratory, ASLab (2008)
30. Hernández, C., Sanz, R., López, I.: Consciousness in cognitive architectures: a principled analysis of RCS, Soar and ACT–R. Technical Report R–2008–004, Autonomous Systems Laboratory, ASLab (2008)
31. Hesse, W.: Ontologies in the software engineering process. In: Lenz, R. (ed.) Tagungsband Workshop on Enterprise Application Integration (EAI 2005), GITO–Verlag, Berlin (2005)
32. Hruby, P.: Ontology–based domain–driven design. In: Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2005), San Diego, California, U.S.A (2005)
33. IEEE: IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society, New York, IEEE std 610.12 1990 edition (1990)
34. IEEE: IEEE Recommended Practice for Architectural Description for Software- Intensive Systems. Institute for Electrical and Electronics Engineering, New York, IEEE std 1471-2000 edition (2000)
35. Keet, C., Artale, A.: Representing and reasoning over a taxonomy of part–whole relations. Applied Ontology (1), 1–17 (2007)
36. Klir, G.J.: Approach to General Systems Theory. Van Norstrand Reinhold, New York (1969)
37. Klir, G.J.: Facets of Systems Science. Plenum Press (1991)
38. Klir, G.J., Elias, D.: Architecture of Systems Problem Solving. IFSR International Series on Systems Science and Engineering, vol. 21. Kluwer Academic Publishers (2003)
39. Klir, G.J.K.: The emergence of two–dimensional science in the information society. Systems Research 2(1), 33–41 (1985)
40. Lehtihet, E., Strassner, J., Agoulmine, N., Foghlú, M.Ó.: Ontology-Based Knowledge Representation for Self-governing Systems. In: State, R., van der Meer, S., O'Sullivan, D., Pfeifer, T. (eds.) DSOM 2006. LNCS, vol. 4269, pp. 74–85. Springer, Heidelberg (2006)
41. López, I.: A Foundation for Perception in Autonomous Systems. Phd thesis, Universidad Politécnica de Madrid (2007)
42. Malucelli, A., Palzer, D., Oliveira, E.: 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), Tokyo, Japan, pp. 26–35. IEEE Computer Society (2005)
43. Mizoguchi, R.: Tutorial on ontological engineering - part 2: ontology development, tools and languages. New Generation Computing 22(1), 61–96 (2004)
44. Morbach, J., Bayer, B., Wiesner, A., Yang, A., Marquardt, W.: OntoCAPE 2.0: the upper level. Technical Report LPT–2008–25, RWTH Aachen University (2008)
45. Morbach, J., Wiesner, A., Marquardt, W.: A meta model for the design of domain ontologies. Technical Report LPT-2008-24, RWTH Aachen University (2007)
46. Noy, N., McGuinness, D.: Ontology development 101: A guide to creating your first ontology. Technical Report KSL–01–05, Stanford Knowledge Systems Laboratory (2001)
47. OMG: MDA Guide Version 1.0.1. Object Management Group (2003)
48. Pinto, H.S., Martins, J.P.: Ontologies: How can they be built. Knowledge and Information Systems 6(4), 441–464 (2004)
49. OMG: OMG SysML Specification. Object Management Group, v 1.1 edition (2008)

50. OMG: Software and systems process engineering meta–model specification version 2.0. OMG Formal Specification 2008–04–01, Object Management Group, Inc. (2008)
51. OMG: OMG Unified Modeling Language (OMG UML) Infrastructure and Superstructure Version 2.2 (2009)
52. OMG: Ontology Definition Metamodel Version 1.0. Object Management Group (2009)
53. Provine, R., Uschold, M., Smith, S.: Observations on the use of ontologies for autonomous vehicle navigation planning. In: 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems, Stanford, California (2004)
54. Ruiz, F., Hilera, J.: Ontologies for Software Engineering and Software Technology. Springer, Heidelberg (2006)
55. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual, 2nd edn. Object Technology. Addison-Wesley (2004)
56. Sanz, R., Hernández, C., Gómez, J., Bermejo-Alonso, J., Rodríguez, M., Hernando, A., Sánchez, G.: Systems, models and self–awareness: towards architectural models of consciousness. International Journal of Machine Consciousness 1(2), 255–279 (2009)
57. Sanz, R., Hernández, C., Rodríguez, M.: ASys models: Model–driven engineering in ASys. Technical Report R–2007–016, Autonomous Systems Laboratory, ASLab (2007)
58. Sanz, R., López, I., Bermejo, J.: A rationale and vision for machine consciousness in complex controllers. In: Artificial Consciousness, pp. 141–155. Imprint Academic, Exeter (2007)
59. Sanz, R., López, I., Bermejo, J., Chinchilla, R., Conde, R.: Self-X: The control within. In: 16th IFAC World Congress. IFAC, Praga (2005)
60. Sanz, R., Matia, F., Puente, E.A.: The ICa approach to intelligent autonomous systems. In: Microprocessor–based and Intelligent Systems Engineering. Kluwer Academic Publishers (1999)
61. Sanz, R., Rodríguez, M.: The ASys vision: Engineering any-x autonomous system. Technical Report R-2007-001, Autonomous Systems Laboratory, ASLab (2008)
62. Schlenoff, C., Messina, E.: A robot ontology for urban search and rescue. In: 2005 ACM Workshop on Research in Knowledge Representation for Autonomous Systems, pp. 27–34. ACM Press, Budapest (2005)
63. Scrapper, C., Balakirsky, S.: Knowledge representation for on–road driving. In: 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems, Stanford, California (2004)
64. Segarra, M.J.: CORBA control systems. PhD thesis, Universidad Politécnica de Madrid (2005)
65. Stahl, T., Völter, M.: Model-Driven Software Development: technology, engineering, management. John Wiley and Sons, Ltd. (2006)
66. Stojanovic, L., Abecker, A., Stojanovic, N., Studer, R.: Ontology–based correlation engines. In: International Conference on Autonomic Computing (ICAC 2004), pp. 304–305 (2004a)
67. Stojanovic, L., Schneider, J., Maedche, A., Libischer, S., Studer, R., Lumpp, T., Abecker, A., Breiter, G., Dinger, J.: The role of ontologies in autonomic computing systems. IBM Systems Journal 43(3), 598–616 (2004b)
68. Tamma, V., Cranefield, S., Finin, T., Willmott, S.: Ontologies for Agents: Theory and Experiences. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkhäuser (2005)
69. Tziallas, G., Theodoulidis, B.: Building autonomic computing systems based on ontological component models and a controller synthesis algorithm. In: 14th International Workshop on Database and Expert Systems Applications (DEXA 2003), pp. 674–680, Prague, Czech Republic (2003)
70. University of Toronto: GRL ontology (2004)
71. UPM-ICEA-Team: Case studies of perception and system analysis. Technical Report ASLab-ICEA-R-2006-015, 1.0 Final, Autonomous Systems Laboratory (ASLab) (2006)

72. UPM-ICEA-Team: ICEA glossary: integration, cognition, emotion, autonomy. Technical Report ASLab-ICEA-R-2006-014, Autonomous Systems Laboratory (ASLab) (2006)
73. UPM-ICEA-Team: A vision of general autonomous systems. Technical Report ASLab-ICEA-R-2006-018, 1.0 Final, Autonomous Systems Laboratory (ASLab) (2006)
74. UPM-ICEA-Team: A vision of perception in autonomous systems. Technical Report ASLab-ICEA-R-2006-017, Autonomous Systems Laboratory (ASLab) (2006)
75. Uschold, M., King, M.: Towards a methodology for builiding ontologies. In: Skuce, D. (ed.) IJCAI 1995 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, pp. 6.1–6.10 (1995)
76. Uschold, M., Provine, R., Smith, S., Schlenoff, C., Balikirsky, S.: Ontologies for world modeling in autonomous vehicles. In: 18th International Joint Conference on Artificial Intelligence, IJCAI 2003 (2003)
77. van Lamsweerde, A.: From System Goals to Software Architecture. In: Bernardo, M., Inverardi, P. (eds.) SFM 2003. LNCS, vol. 2804, pp. 25–43. Springer, Heidelberg (2003)
78. van Lamsweerde, A.: Requirements engineering: From craft to discipline. In: FSE 2008: 16th ACM Sigsoft International Symposium on the Foundations of Software Engineering, Atlanta, U.S.A (2008)
79. Wongthongtham, P., Chang, E., Dillon, T.: Towards ontology-based software engineering for multi-site software development. In: 3rd IEEE International Conference on Industrial Informatics (INDIN), Perth, Australia, pp. 362–365 (2005)
80. Yu, E.: Towards modelling and reasoning support for early–phase requirements engineering. In: 3rd IEEE International Symposium on Requirements Engineering (RE 1997), Washington, D.C., USA, pp. 226–235 (1997)