Ian David Lockhart Bogle and Michael Fairweather (Editors), Proceedings of the 22nd European Symposium on Computer Aided Process Engineering, 17 - 20 June 2012, London.

Ontology-driven description and engineering of Autonomous Systems: application to process systems engineering

M. Rodríguez (manuel.rodriguezh@upm.es), J. Bermejo-Alonso, C. Hernandez Corbato, R. Sanz

Abstract

Autonomous systems refer to systems capable of operating in a real world environment without any form of external control for extended periods of time. Autonomy is a desired goal for every system as it improves its performance, safety and profit. Ontologies are a way to conceptualize the knowledge of a specific domain. In this paper an ontology for the description of autonomous systems as well as for its development (engineering) is presented and applied to a process. This ontology is intended to be applied and used to generate final applications following a model driven methodology.

Keywords: Ontology, Systems engineering, Autonomous Systems

1. Introduction

Autonomous systems refer to systems capable of operating in a real world environment without any form of external control for extended periods of time. Reasons to provide systems with autonomy range from cost reduction to improved performance and dependability. Moreover, managing the increasing complexity of these systems has turned into delegating their configuration, optimisation, and repair to the systems themselves. In the process industry systems are more complex every day because they are highly interrelated and integrated, not only process operation units but the control and external applications (including the software that supports them). The use of a ontology on an existing system has clear benefits (as reuse, reasoning, documentation, etc.) but what is missing is a way to engineer that complex system to comply with a certain set of requirements, in this case including autonomy. The ontology presented in this paper addresses both tasks at the same time, the engineering and the description of the system. The remaining of the paper is structured as follows: section 2 presents the requirements of the ontology, sections 3 and 4 describe the ontology and its methodology, section 5 shows an application and, finally, section 6 draws conclusions.

2. Ontology for Autonomous Systems Requirements

The ontology for autonomous systems (OASys) needed to address two domains. On the one hand, the domain of Autonomous Systems, as the ontology will be used to describe an autonomous system's structure, function, and behaviour. On the other hand, the domain of software engineering as the ontology will also describe the autonomous system's engineering process.

Different requirements were considered for OASys. Design decisions and further analysis of the requirements and their final implementation were also carried out

[Bermejo-Alonso2011]. This section describes such requirements, as well as their implementation in the actual features of the ontology.

The first element considered was the *purpose* of the ontology, as establishing its aim and future use aids at defining with detail the ontological elements. As such, OASys defines the concepts and constructs for the definition of the autonomous systems. The ontology also captures the concepts to support the engineering process of this kind of systems. The *type of ontology* [GomezPerez2004] is a domain ontology.

An additional requirement was to define its *general structure* according to its purpose. Hence, the structure needed to address both general and autonomous systems knowledge, as well as providing ontological elements both for the autonomous system's engineering process. OASys was organised in two orthogonal dimensions: different levels of abstraction to separate general knowledge from particular one; the second one focusing on the separation between the autonomous system description and its engineering process.

To cater for the requirement of *extendibility*, concepts were classified using subontologies and packages. Subontologies grouped ontological elements closely related to a domain viewpoint. Packages classified the ontological constructs within a particular subontology according to a concrete aspect. Both organisational elements allow modifications of OASys without having to change the entire ontology.

Finally, attention was paid to the methodology and the implementation. From the available analysed methodologies, METHONTOLOGY [LopezFernandez1997] (along with additional guidelines [Mizoguchi2004]) was chosen since it describes the development process with enough level of details but at the same time, allowing for flexibility.

Additionally, the analysis of available techniques, languages and tools was conducted to establish the language to be used for implementing OASys. As a result, UML [OMG2009] was chosen to specify the ontology. It's been used in ontologies for autonomous systems and software engineering [Gasevic2006].

3. OASys description

This section describes the ontologies, subontologies and packages considered in OASys [Bermejo-Alonso2010]. Each package has been developed from the package's purpose to its final UML formalisation. OASys was formalised as two main ontologies: the ASys Ontology, and the ASys Engineering Ontology. Both ontologies were conceived to be complementary to each other, with the latter constructing and guiding the use of the contents of the first one during the autonomous system's conceptual modelling in the OASys-based Engineering Methodology.

3.1. ASys Ontology

ASys Ontology gathers, at different levels of abstraction, the ontological constructs necessary to describe and characterise an autonomous system, as two subontologies: the System Subontology, and the ASys Subontology. At the upper level of abstraction, the System Subontology contains different packages gathering the concepts for any system's definition: General Systems Theory concepts, Mereology, and Topology. At a lower level of abstraction, the ASys Subontology consists of several packages to address an autonomous system description: the Perception, the Knowledge, the Thought, the Action, and the Device packages. The ontological elements in these packages refine and specialise the concepts and relationships defined in the System Subontology.

3.2. ASys Engineering Ontology

The ASys Engineering Ontology provides autonomous system's engineering ontological constructs, organised in two subontologies: the System Engineering Subontology, and the ASys Engineering Subontology. The System Engineering Subontology gathers the concepts related to any system engineering process as general as possible based on system's engineering and software engineering methodologies, organised into the Requirement, the System Perspective, the Engineering Process and the Model–driven packages. As specialisation for autonomous systems, the ASys Engineering Subontology contains the ontological constructs to describe an autonomous system's engineering process. It is internally organised into different packages: the ASys Requirement, the ASys Perspective, and the ASys Engineering Process packages.

4. OASys based methodology

The OASys–based Engineering Methodology is an ontology-based autonomous system generic development process based on the OASys ontological constructs [Bermejo-Alonso2010b], in terms of phases, tasks and work products, having as guideline the ontological elements in the System Engineering and ASys Engineering subontologies. As a result for the different phases and tasks, conceptual models of the considered autonomous systems are obtained as UML diagrams.

The methodology consists of two main phases, ASys Requirement to identify the autonomous system's requirements, and ASys Analysis to consider the autonomous system's analysis of its structure, behaviour and function. There might be an initial ASys Views phase, to identify those engineering views of interest in the autonomous system development. The ASys Requirement phase identifies and elicits stakeholders' requirements for the autonomous system and subsystems. The ASys Analysis phase describes the autonomous system from different viewpoints, considering different tasks such as Structural Analysis, Behavioural Analysis and Functional Analysis.

5. OASys Application

The presented methodology has been applied to an existing pilot plant, called Process Control Testbed (PCT). Its main component is a Jacketed Continuous Stirred Tank Reactor (JCSTR), as well as the related instrumentation and control system. The aim is to provide the system with cognitive capabilities to carry out complex tasks such as fault diagnosis, alarm management, and control system reconfiguration. Figure 1 shows the P&I diagram of this process.

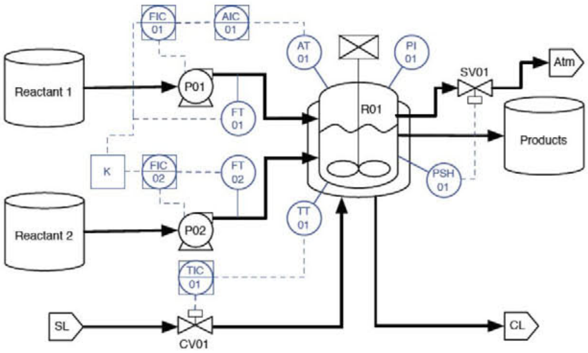

Figure 1. Process Control Testbed P&I

There are two main phases in the methodology, the first one, the ASys Requirements phase, identifies the needed conditions for the system and it establishes the different system's use cases. Figure 2 shows the PCT system use case "Start Plant". Every use case is detailed following a pattern that keeps all the relevant information like name, source, description, functional focus, actors, basic events, preconditions and postconditions. The second step is the ASys Analysis phase which comprises the structural as well as the functional analysis. The Structural Analysis has as objective the analysis of a system considering its structural aspects, under a *Structural Viewpoint*. Different *Engineering Models* in the form of *Structural Models* can be obtained as result of performing the *System Modeling* Subtask defined in the ASys Engineering Package. The *Structural Model* is a model kind to describe an autonomous system from a structural viewpoint that conforms, as a matter of fact, to a specific level of detail that could be further refined.

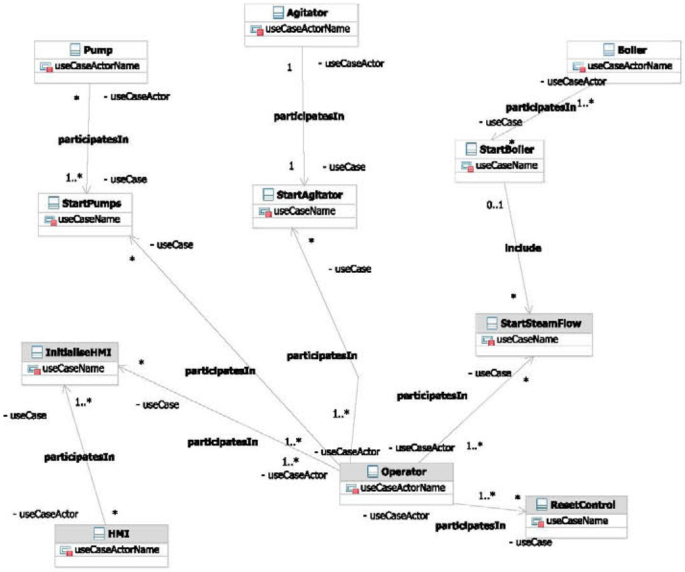

Figure 2. Start Plant Use Case Diagram

For the PCT, the Structural Model consists of different *Structure Models* to describe the elements, as well as the *Topology Models* to describe topological connections among the components in the system. The *Structure Model* specialises the ontological elements in the General Systems Theory and Mereology Packages. The Process Control Testbed systems (CSTR, Hydraulics, Control and Instrumentation) are further decomposed in smaller systems (for example the Hydraulics has tubing, pump, valves, which decompose in control and safety valves). In the model, the relationships between the System (the PCT), and the Subsystems (CSTR, Hydraulics, etc.) are expressed by means of the UML composition association. Finally, the functional analysis depicts the function of every described component in the structural model. A behavioural analysis is also included in order to differentiate between the intended behaviour, as expressed in the functional analysis, from the feasible behaviour. This last analysis is especially useful to identify and diagnose faults in the system under consideration. All the views (requirements, structural and functional) are related to a physical component that can be browsed and identified in all of the views having a complete description of what it is, what it does and what are its requirements.

6. Conclusions

This paper has presented a model-based approach to develop autonomous systems. To obtain the conceptual models, a framework consisting of an ontology and a methodology has been defined and developed. The methodology defines the process for the characterisation and the engineering, specifying the phases, tasks, and conceptual models to be obtained, using as underlying semantic support the ontological elements defined in OASys. The approach here has been to define the different elements to describe the system structure and function in a way general enough to be reused among different applications. The obtained models are the same ones used by the engineers to build the system. This will ultimately provide the system with self-engineering capabilities required for robust autonomy. The application of the ontology and the methodology has been shown on a process system.

References

J. Bermejo-Alonso, R. Sanz, M. Rodríguez, and C. Hernández, 2011, Engineering an Ontology for Autonomous Systems. Proc. the Intl. Conf. on Knowledge Eng. and Ontology Devpt pp 47-58.

J. Bermejo-Alonso, 2010, OASys: Ontology for Autonomous Systems, PhD dissertation, E.T.S.I.I.M., U.P.M

D. Gasevic, D. Djuric, and V. Devedzic, 2006, Model-Driven Architecture and Ontology Development, Springer-Verlag Berlin Heidelberg.

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

M. López Fernández, A. Gómez Pérez, and N. Juristo, 1997, METHONTOLOGY: from ontological art towards ontological engineering, Proceedings of the AAAI 97 Spring Symposium on Ontological Engineering, Stanford University, CA, U.S.A, Pages 33-40.

R. Mizoguchi, 2004, Tutorial on ontological engineering - part 2: ontology development, tools and languages, New Generation Computing, Vol. 22, No. 1, OhmSha & Springer, Pages 61-96.

Object Managment Group, 2009, Unified Modeling Languae Infrastructure and Structure v2.2.

J. Bermejo-Alonso, R. Sanz, M. Rodríguez, and C. Hernández, 2010b, An Ontological Framework for Autonomous Systems Modelling. Int Journal On Advances in Intelligent Systems , Vol 3 and 4, pages 211-225.