Towards holonic multiagent systems: Ontology for supervision tool boxes

Silvia Suárez, Beatriz López, Joaquim Melendez

Universitat de Girona.
Campus Montilivi EPS Building P4, Girona, Spain
{sasuarez, blopez, quimmel}@eia.udg.es

Abstract. During years of research in control field, some supervision, detection and diagnosis techniques have been developed. Now, advances on distributed and ubiquitous computing, networking and sensors provide new environments to integrate control techniques. However, this integration is not straigforward, reason of why new methods and approaches are required. In this sense, holons and agents seems a promising parading in which future control systems can be developed. The work presented in this paper is framed on this new paradigm. Particularly, we propose an ontology for dealing with control tool boxes integration. The ontology has been elaborated with Protegé2000.

1. Introduction

Distributed Control System (DCS) extensively used fifteen years ago in the process industry is evolving towards a higher level solutions based on a better connectivity among applications and process that assures data flowing from process to manage boards. Actual requirements of flexibility, traceability and quality force the permanent communication among all *agents* (suppliers, factories, vendors, maintenance, etc.) that participate in the final product. Next figure shows the basic architecture of a SCADA (Supervisory Control and Data Acquisition) software. They are clearly oriented to guarantee the integration between process (Instrumentation Communication Interface), operators (HMI), supervisors (SPC/SQC), manager and other enterprise resources (ERP).

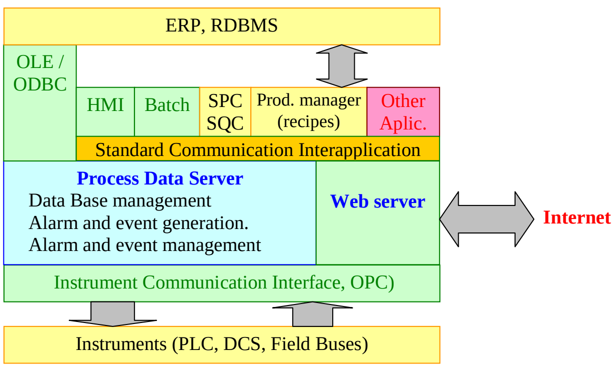

Fig. 1: Basic structure of a SCADA system (from [20]).

However, the integration of control techniques rises on the main challenge of shifting from traditional control systems as processes with single controllers to control systems as collections of heterogeneous physical and information systems, with complex inter-connexions and interactions [1]. New methods and approaches are then required, and holons and agents, are promising paradigms.

On one hand, a holon is “an autonomous and co-operative building block of a manufacturing system for transforming, transporting, storing physical and information objects [2], [3]. It consists on a control part and an optional physical processing part (see figure 2). A holon can itself also consist of other holons [3]. This conception of holons is clearly an extension of actual SCADA systems (See figure 1) with improved communication and processing capabilities clearly oriented to decision making.

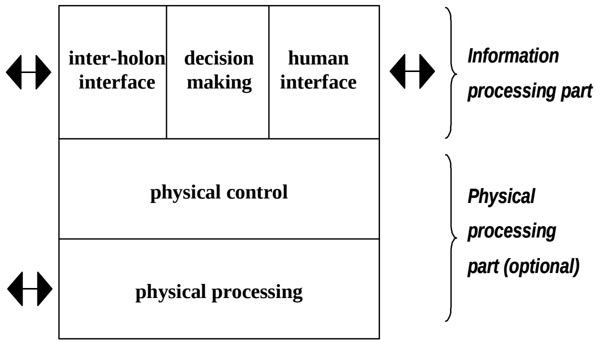

Fig. 2: General architecture of a holon (from [3]).

On the other hand, agents provide autonomy with respect to the system capacity to a given environment [4]. Agent Technology, although broadly extended on open applications as Internet services, has been recently introduced on the control field [5], [6], [7] and [8].

Although both approaches, holons and agents, share many basic concepts, research in each area has been developed independently for the most part: holonic systems research has focused on manufacturing systems, and Agent Technology research on the development of interconnected systems in which data, control, expertise or resources are distributed. Being aware of the common interest, there are recent efforts to understand whether holons and agents are depth or not (see [9] for a comparative analysis) and to joint both communities [10], [11], [12], [13], [14]. This has lead to form the term Holonic Multiagent Systems, a novel paradigm for managing, modelling and supporting complex systems [10], [14]. This new paradigm provides two main benefits. On one hand, holons provides soundness and robustness, typical characteristics of the system engineering developments. On the hand, agents facilitate the integration of heterogeneous systems.

Our work is in line with this new approach to integrate heterogeneous control systems. More that building new control techniques, we focus on the integration of them. As a first step, we have participated in the development of tool boxes that encapsulates and describes control techniques [15]. Now, we are dealing with the problem of making tool boxes interconnection operational. In such challenge, one of the main drawbacks consists on information sharing, and so, ontologies play an essential role. Ontologies have established themselves as a powerful tool to enable knowledge sharing, and achieve semantic interoperability among heterogeneous, distributed agent systems [16]. Our ultimate goal is to use the ontology to integrate control tool boxes in a holonic multiagent system.

This paper is organized as follows: First, in section 2, we detail some related work. Then, in section 3, we provide our ontology. We end in section 4 by giving some conclusions.

2. Related works

There are some previous works on the definition of ontologies for diagnosis and fault detection. In [17], an ontology for fault diagnosis in Electrical Networks is presented. This paper remarks the clear need of a standard representation of the electrical network. In the past, each system has been build using a different representation of the network. However, the domain knowledge was the same in all cases. The works in this area are aimed at the representation of this knowledge in a way that is reusable by several applications. Here, the authors say that the obvious way to identify what is reusable and how to represent it is to look at the domain knowledge required in applications where the knowledge about the electrical network is crucial, e.g. fault diagnosis. The long term objective is to obtain a standard representation for the applications and electrical utilities. This paper is similar to our approach, since the authors aim's is to develop a standard ontological study. However, they are constrained to a domain (electrical), while we are concerned on control techniques, whatever domain it is.

In addition, in [18] an ontological analysis of fault process is presented in a general way. This study ontologically analyzes the fault process, i.e., causal chains in which the faults and the symptoms are induced, aiming at articulation of concepts which can categorize the fault processes and the faults. The ontology of faults provides a conceptual vocabulary to explicate the scope of a diagnostic activity performed by a reasoning mechanism using a kind of model. The objective of this paper is to identify richer vocabulary including such concepts from the viewpoints not of logic but of ontology of physical systems. Some questions are proposed by the authors, i.e.: How the observed symptom is induced by the initial fault, what type of phenomena appears in the fault process, and what categories of the deeper causes of faults exist. Our approach is broader than [18], due to the fact that our ultimate goal is to build an ontology for all the steps involved in control systems: supervision, fault detection and diagnosis.

On the other hand, in [19] a representation of fault cases for supporting fault diagnosis tasks is explained. The authors use an ontology for Representation of fault case information. The specified terms in this approach are limited to fault cases. This approach is different to our approach but it helps to observe and know the components of the fault cases presented.

3. Elaborated ontology

In order to elaborate an ontology for control tasks, we have used the terms proposed in [20] and [21] by the area of supervision, fault detection and diagnosis. Therefore, main terms are organized in variables, systems behaviours, supervisory tasks, models and system properties (see Figure 3). Each term is defined in properties and relations, generating a complex network of classes, subclasses, instances and slots. Following we describe all the terms.

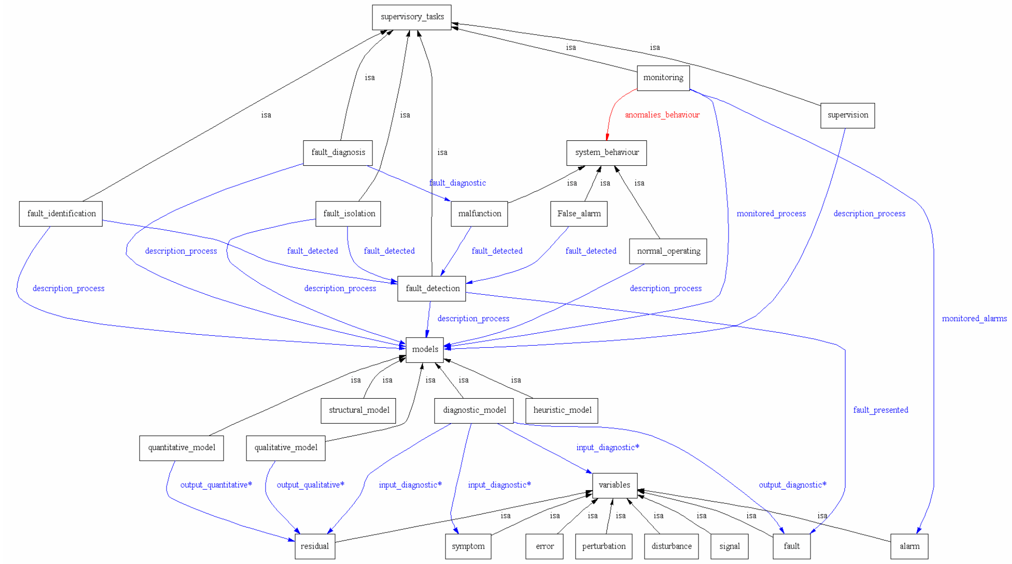

Fig. 3: Ontology general diagram

3.1 Variables

- Signals are defined as physical measures or perceptible variables that communicate information and messages. Attributes: Units, Range, Typology (quantitative/qualitative, continuous/sampled/discrete events).
- Fault: Unpermitted deviation of at least one characteristic property or parameter of the system from acceptable / usual / standard condition. It is composed of the following attributes: cause, duration, final_time, typology_of_fault (intermitent/permanent, evolutive/abrupt, additive/parametric), location_fault, Descriptor_faul, Size_fault, starting_time.
- Error: Deviation between a computed variable (typically and output or state variable) and the true, specified or theoretically correct value. It presents the following attributes: two inputs (to compute it): correct_value, duration, final_time, measured_computed_value, result, starting_time
- Disturbance: An unknown (and uncontrolled) input acting on a system. The attributes of this subclass are: typology (additive /multiplicative, etc), shape, duration final_time, input_point (if known, related with the physical system and /or the structural model), effects and starting_time.
- Perturbation: An input acting on a system which results in a temporary departure from current state. The attributes are: idem disturbance.
- Residual: Fault indicator, based on deviation between measurements and model-equation-based computations. Particular case of error. It has the following attributes: deviation, duration, final_time, result (fault detection, it is decide about the presence or absence of faults), starting_time, decision mechanism, Signature
- Symptom: Change of an observable quantity from normal behaviour. The attributes of this subclass are: referred quantity, duration, final_time, starting_time, shape (trends) and values.

3.2 System Behaviour

System behaviour is a gross description of the system operating conditions. Basic states can be defined:

- Normal Operating: system behaviour is under specifications.
- Faulty or Malfunction: Intermittent irregularity in fulfilment of a systems desired function. It presents the following attributes: periodicity, starting time, final_time, faults (see fault attributes).
- False alarm: The system is operating properly but the supervisory system has detected some misbehaviour.

3.3 Supervisory Tasks

Tasks are the different kind of operations that must be performed in order to supervise a given system. There are seven main kinds of functions:

- Fault detection: Determination of the presence of faults in a system and time of detection. This subclass is composed of the following attributes: fault_presented (yes /not, without specifying additional information) and time_detection.
- Fault isolation: Identification of relevant attributes of faults present: kind, location and time of detection of a fault. Follows fault detection. The attributes of this subclass are: fault_presented and time detection (input attributes given by the fault detector), fault attributes (kind_fault, location_fault, fault_time) are presented as conclusion of this task, it needs information from specific model of the system (structural model, diagnosis model) in order to perform the task..
- Fault identification: Determination of other relevant attributes of faults: the size and timevariant behaviour of a fault. Follows fault isolation. It presents the attributes: fault attributes (size_fault and timevariant_behaviour).
- Fault diagnosis: Determination of kind, size, location and time of detection of a fault. Follows fault detection. Groups fault isolation and identification. The attributes of this subclass are: time detection, fault_presented and diagnostic (kind_fault, location_fault, size_fault and time_detection). It needs structural and/or diagnosis models to perform the task.
- Monitoring: A continuous real time task of determining the conditions of a physical system, by recording information recognising and indicating anomalies of the behaviour. The attributes are: monitored variables, alarms, events, operating_conditions, tunning parameters (thresholds and similars), anomalies_behaviour and information_recongnising.
- Supervision: Monitoring a physical system and taking appropriate actions to maintain the operation in the case of faults. It has the following attributes: diagnostic, actions, decision_system.

3.4 Models

For the purpose of engineering analysis and design, physical systems are usually represented in some mathematical form; this representation is also called the model of the system. The properties of the model reflect the nature of the system, though in many cases the model may just be an approximation of the true system behaviour. In [20], four classes of models are considered (See figure 4):

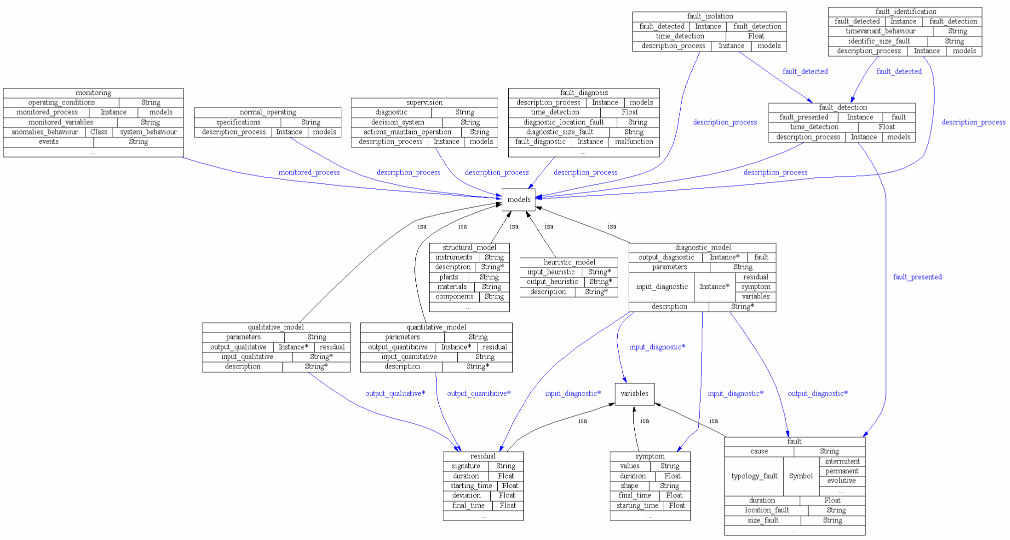

Fig. 4: Ontology particular diagram

- Quantitative model: Use of static and dynamic relations among system variables and parameters in order to describe systems behaviour in quantitative mathematical terms. The attributes of this subclass are: description, input_quantitative, output_quantitative and parameters.
- Qualitative model: Use of static and dynamic relations among system variables and parameters in order to describe systems behaviour in qualitative terms such as causalities or if-then rules. It presents the following attributes: description, input_qualitative, output_qualitative and parameters.
- Diagnostic model: A set of static or dynamic relations which link specific input variables - the symptoms - to specific output variables - the faults. The attributes of this subclass are: description, input_diagnostic (symptoms, residuals, and physical variables), output_diagnostic (fault attributes) and parameters.
- Heuristic_model: The attributes of this subclass are: description, input_heuristic and output_heuristic. Commonly used for diagnosis working with symptoms and attributes of diagnostic model.
- Structural model: Definition of the physical interaction between components, materials and energy sources. Attributes: description, components, (sub)systems, instruments, plants, materials.

3.5 System Properties

Systems properties relate particular characteristics of required for the system. Main properties are: reliability, safety, availability, dependability.

- Reliability: Ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time. Measure: MTBF = Mean Time Between Failure. MTBF = 1\la; la is rate of failure [e.g. failures per year]. It has the following attributes: MTBF, period_time and required_funtion.
- Safety: Ability of a system to not cause a danger for persons, equipment or environment. It presents de following attributes: value_safety.
- Availability: Probability that a system or equipment will operate satisfactorily and effectively at any point of time measure: MTTR: Mean Time To Repair MTTR = 1/µ; µ: rate of repair. The attributes of this class are: MTTR and probability_availability.
- Dependability: A form of availability that has the property of always being available when required. It is the degree of which an item is operable and capable of performing its required function at any randomly chosen time during its specified operating time, provided that the item is available at the start of that period [RAM Dictionary]. It presents the following attributes: degree_dependability and time_dependability.

The ontology has been created using the tool protégé 2000 [22].

4. Conclusions

In this paper we propose an ontology as first step to solve the problem of making operational the integration of several control techniques, so achieving an holistic control system. We have implemented a preliminary version of the ontology on Protege2000. Our next step involves the use of the ontology in a holonic multiagent system, in which several control agents (agentified tool boxes) interact.

Acknowledgments

This work has been supported by the Spanish MCYT project DPI2001-2198.

References

[1] Murray, R.M., Aström, K.J., Boyd, S.P., Brockett, R.W., Stein, G. Future Directions in Control in an Information-Rich World. A summary of the report of the Panel on Future Directions in Control, Dynamics, and Systems. IEEE Control Systems Magazine, April 2003, pp. 20-33.

[2] Christensen, J. Holonic Manufacturing Systems – Initial Architecture and Standards Directions. Proc. First European Conference on Holonic Manufacturing Systems, Hannover, Germany, 1994.

[3] MacFarlane, D., Bussmann, S. Developments in holonic production planning and control. Int. Journal of Production Planning and Control, 11 (6): 522-536, 2000.

[4] Weiss G. (Editor) "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence". The MIT Press, 1999.

[5] Bennett, S. Otto Mayr: Contributions to the History of Feedback Control. IEEE Control Systems Magazine, 22 (2): 29-33, April 2002.

[6] Jennings, N.R., Bussmann, S. Agent-Based Control-Systems. Why They Suited to Engineering Complex Systems. IEEE Control Systems Magazine, June 2003, pp. 61-73

[7] Bussmann, S., Jennings, N.R., Wooldridge, M. On the Identification of Agents in the Design of Production Control Systems, In: P. Ciancarini, M.J. Wooldridge (eds.), Agent-Oriented Software Engineering, LNCS 1957, pp. 141-162. Springer-Verlag: Berlin, Germany, 2001.

[8] Bussmann, S., Schild, K. An Agent-based Approach to the Control of Flexible Production Systems. Proc. 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation (EFTA 2001), pp. 169-174. Antibes Juan-les-pins, France, 2001.

[9] A. Giret, V. Botti. Holons and Agents. Journal of Intelligent Manufacturing. Pdte. publicación, Kluwer, 2004.

[10] HoloMAS 2003, 1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems, September 1-3, 2003, Prague, Czech Republic. http://gerstner.felk.cvut.cz/HoloMAS/2003

[11] Workshop on Holons: Autonomous and Cooperative Agents for Industry, at the 5th International Conference on Autonomous Agents, Montreal, Canada, 29 May 2001. http://isg.enme.ucalgary.ca/workshop.htm

[12] Brennan, R.W.; Fletcher, M.; Norrie, D.H. An agent-based approach to reconfiguration of real-time distributed control systems. Robotics and Automation, IEEE Transactions on , Volume: 18 Issue: 4 , Aug. 2002. Page(s): 444 -451

[13] Fletcher, M.; Brennan, R.W.; Norrie, D.H.; Deadline control in holonic manufacturing using mobile agents. Database and Expert Systems Applications, 2001. Proceedings. 12th International Workshop on , 3-7 Sept. 2001. Page(s): 623 -627

[14] Schillo, M., Fischer, K. Holonic Multiagent Systems. In: Zeitschrift für kÑunstline, n. 3 (to appear).

[15] CHEM project. Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes. http://www.chem-dss.org/

[16] Working group on Ontologies. Agenticies.net. (http://www.csc.liv.ac.uk/~valli/WG-Ontology/index.html)

[17] Bernaras A., Laregoiti I., Bartolomé N. And Corera J. An Ontology for Fault Diagnosis in Electrical Networks. Intelligent Systems Applications to Power Systems, 1996. Proceedings, ISAP '96., International Conference on , 28 Jan-2 Feb, 1996.

[18] Kitamura Yoshinobu and Mizoguchi Riichiro, An Ontological Analysis of Fault Process and Category of Faults. Proc. Of Tenth Interantional Workshop on Principles of Diagnosis (DX-99), pp.118-128, June 8-11, 1999.

[19] Kato, Yoshikiyo, Shirakawa Takahiro and Hori Koichi. Utilizing Fault Cases for Supporting Fault Diagnosis Tasks.

[20] Isermann and Ballé. “Trends in the application of model based fault detection and diagnosis of technical processes”, in IFAC -13 th Triennial World Congress, ref. 7f-01 1, San Francisco, USA. http://w3.rt.e-technik.tu-darmstadt.de/institut/terminology.en.html

[21] Colomer J, Meléndez J. “Sistemas de Supervisión”. Cuadernos CEA-IFAC. Ed. CETISA, 2000.

[22] Protegé 2000. http://protege.stanford.edu/