Moving Integrated Product Development to Service Clouds in the Global Economy
J. Cha et al. (Eds.)

doi:10.3233/978-1-61499-440-4-547

Towards Self-evolutionary Cyber Physical Systems

Sudeep GHIMIRE[1,2], Fernando LUIS-FERRIERA[1,2], Ricardo JARDIM-GONCALVES[1,2], Tahereh NODEHI[1]
[1] Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[2] Centre of Technology and Systems, CTS, UNINOVA, 2829-516 Caparica, Portugal
{sud, flf, rg}@uninova.pt, t.nodehi@campus.fct.unl.pt

Abstract. Cyber-Physical Systems (CPS) are systems that comprises of integration of computation and physical processes. In CPS, computation and communication involves interaction with physical environment to add new capabilities and characteristics to systems. However, the integration of data acquisition systems from physical environment increases the factor for unreliability to the overall system because of the unpredictable behaviors of the physical world. So, it's a vital research challenge to address the design of CPS, which is adaptive to the changes that will occur at runtime based on the uncertainties in the physical world. At the same time making use of the knowledge acquired during operation for is an important aspect for scalable evolutionary systems. This poses an interesting research paradigm for design of CPS to capture process and integrate knowledge from physical world into the digital systems, by taking into consideration of CPS domain specific semantics. Major challenge addressed in this research work is to pave a path towards understanding of evolutionary systems along with reference architecture for evolutionary cyber physical systems.

Keywords: Cyber Physical System, Adaptive Systems, Self-Evolutionary systems

Introduction

Cyber Physical Systems (CPS) are systems that comprises of integration of computation and physical processes. CPS comprises of embedded computing devices and networks that monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. Interaction between computational components and physical environment is research challenge in the design and implementation of CPS. At one hand, interaction with physical environment will provide added value with new capabilities and characteristics to systems, while in the other hand inclusion of physical processes not only increases the complexity of the system but also increases the uncertainties in the behavior of the system.

Since, we are building system which takes into account of the external stimuli, it is obvious that the computation behavior of the system is not only dependent on the internal configuration but is guided by the parameters from the physical world. These external factors may either reflect an adaptive mode in which changes of environment determine changes in the computational system, or may be determined by internal imperatives in which a monitoring process examines the computational behavior of a system and determines appropriate evolutionary actions when the system requires modification. This paves the path for the self-evolutionary cyber physical systems (eCPS). This research work thus provides an insight on evolutionary CPS with a study on the different domains that can be a strong base on the study of eCPS. We also provide a reference architecture that is being used for realization of an instance of eCPS within the scope of the research we tend to continue.

1. Background and Related work

Flexible Automatic Assembly (FAA) cells or Hyper Flexible Automatic Assembly, as defined in [1] and self-adaptive systems as discussed in [2] provide the foundation for study of evolutionary systems. In these methodologies, the evolution is handled manually. In the context of eCPS the system must be self-evolving i.e. automatically detect the changes in external circumstances or internal conditions and be prepared to handle the changes. eCPS should be capable of selecting alternate software modules and/or be able to dynamically swap these modules while keeping the consistency over the functionality and performance. One of the most interesting previous work is Re-configurable Assembly Systems (RAS) which focused on the principle that innovative product design must not be limited by assembly process constraints, which are interestingly discussed in [3] and [4].

In the domain of manufacturing systems, Holonic Manufacturing Systems (HMS) [5] described a method by which systems could be sub-divided into entities with selected functionality (holons), and subsequently introduced distributed control approaches, as proven by [6]. However, apart from the distributed control approach, HMS does not represent evolvability as the holons are standard manufacturing system components (but not process-oriented), and it does not describe any evolutionary qualities. Another interesting work is presented in [7] where the author sketches rough taxonomy of self-organization which is of relevance in the study of cognitive and biological systems to provide the semiotics of Evolutionary Systems.

1.1. Concepts of Evolutionary Systems

CPS deal with physical environment which tends to be time and context varying with the dynamics of the environment thus incorporating evolutionary behavior. eCPS can be achieved by following the principles discussed in [2] and [8] viz. 'no hard constraints imposition on the system thus ensuing fully independent process selection and execution based on the real-time requirements' and 'need to have an inherent capability to address the new or changing set of requirements, by using the knowledge acquired over time'.

For the realization of eCPS, design and implementation decisions and objectives are to be set at architectural or higher level using exploited protocols, standards or specifications. At the same time the integration of knowledge representation system to capture the domain of run time environment and rules that address the operation of the overall process and their ontological relations are important to achieve eCPS.

Blueprints or templates based system requirements, design and implementation provides the necessary flexibility that the eCPS requires. Based on these discussions, we can outline the following characteristics for eCPS:

- The systems need to be fully “reconfigurable” based on platforms that can exhibit an emergent behavior.
- The reconfigurable system has to be designed by the composition of process oriented components (with high level of abstraction) so as to achieve higher level of granularity over emergent behavior.
- The functional set of the components should be represented by using formal language, which is machine readable, so that the system can automatically determine the functionality of the components, map them with the domain knowledge and use/discard the component as the system evolves.
- The change in paradigm is not addressed on the code level, but at higher level of abstraction, so that the run-time behavior of the system is not hard coded but defined by establishing relationship between components, their interactions and interfaces.

To achieve these characteristics, a number of principles, proposed in the EUPASS [9] and IDEAS projects [10] can be adopted for the paradigm of eCPS. The major objectives of eCPS are thus:

1. Optimized orchestration: The interaction between the components comprising the overall system needs to be defined and implemented in an agile fashion. This can be achieved by adopting the principles from multi-agent based, distributed control approach with embedded controllers. So that they can act as independently and interact with other components without any inferences.
2. Adaptability: Modularity between components allows for stepwise upgradeability and flexibility to changes. The actual system may also adapt to minor changes via its control system, which, being skill-based, allows for emergent behavior to be exploited in the global scenario.
3. Robustness: Each of the components of the overall system are dedicated, small, independent and have their own control systems. The control system is goal-oriented, and the system is process-oriented. This results in a dedicated system based on an adaptable concept with advanced interfaces.
4. Plugability: The components of the overall system have to be pluggable into the main work-flow of the eCPS at run time. This can be achieved by designing element which are very task-specific in order to accomplish only a simple action with well-defined interfaces for accessing the providing functionalities. The overall system is achieved by complex tasks as the union of several simple tasks.

Fundamentally, the discussion in this section suggests that eCPS can be achieved if the lowest building blocks of a system are those that exhibit the highest rate of adaptability and/or evolvability.

2. Towards Self- evolutionary CPSs: A Research Roadmap

As discussed in the previous sections to achieve eCPS, the modular design and implementation methodologies are very important i.e. each of the smallest possible component have well defined capabilities and interaction with other components. The main issue to be addressed in this section is describing the areas in which eCPS control systems are getting inspiration to solve the requirements at fine granularity. Numerous scientific domains investigating phenomena which eCPS also can provide helpful tools and valuable background to cope with the complexity of manufacturing systems [11]. Some important research domains from which the principles and results can be adopted for the realization of eCPS are:

- Complexity Theory

Complexity Theory looks for simple causes leading to complex behaviors [12]. All CPS inherit the characteristics of complex systems i.e. they are spatially and/or temporally extended non-linear systems. eCPS is composed of numerous simple modules which are connected to each other and have multi-lateral interactions. The independent components have high degree of freedom, yet in the run-time environment they have to function within the global constraints to achieve the system functionalities.

- Natural life and Artificial Life

Evolution has been long studied in the natural life by many prominent biologists. The principles and theories from natural life are useful to create life-like behaviors with the capability of evolution on computers and other “artificial” media. eCPS are very similar to artificial living systems which have modifiable structure and exhibit self-organization, adapt to their environment and react to stimuli. They are capable of evolving according to the circumstances, namely in terms of state of available resources and formulate new configuration to address the changes in the overall paradigm. Emergent behavior of an ecosystem of complex computational systems is studied in [13] and provides an interesting discussion on works that have been inspired from natural sciences.

- Autonomic Computing

Autonomic Computing is and important research domain that provides fundamental concept for eCPS. The vision of Autonomic [12] refers to the tendency of computers to become ubiquitous. In the scenario of complex systems like the case of CPS, it tends to form large network of computing devices and smart objects with complex and multiple interactions, leading to increasing difficulty to manage. Thus, autonomic computing in general inherits the necessity of systems that will be able to take of itself with minimized user interactions and need for reprogramming. Even though each components of eCPS (or autonomous systems) are modules of fine granularity with self-computational power, it is necessary to find new ways of coordination and automatic plugability, which is exactly what eCPS want to address.

- Multi-Agent Systems

In the paradigm of CPS, the presence of wider aspect of contextual information, the units interacting with the system need to be modeled as agents which can be a human, an association, an animal, or a piece of software, sensors or network of sensors, tagged objects etc. which are eventually connected to some computational components. The fundamental characteristics of such agents are identity, intelligence and the ability to act and react in order to persecute goals. Agents have at least a certain degree of autonomy and can compete or collaborate with others, which will provide the foundation for reconfigurable orchestration to meet the needs in the change in context. [14] provides an interesting interlink between evolutionary systems and multi-agent systems.

- Semiotics

Semiotics concerns the study of signs/symbols in three basic dimensions: syntactic (rule-based operations between signs within the sign system), semantics (relationship between signs and the world external to the sign system), and pragmatics (evaluation of the sign system regarding the goals of their users) [15]. This principle, even though originates from the biological science provides strong theoretical foundation that can be adopted in the area of eCPS. [7] provides and interesting discussion on interlink between semiotics and AI, which will be an interesting foundation for eCPS.

3. Methodology

In this section we will provide a reference architecture that will be adopted for the realization of the eCPS. But before we will take into account of the reference architecture, this section explains the overall cycle of eCPS. Figure 1shows the overall processes in the lifecycle of the runtime environment of eCPS. The phases are basically divided into monitoring and execution planes. Monitoring phase has a continuous collection of data from the physical world and also the changes in system requirements if any, that reflects the current state of the overall system. The eCPS reference architecture should provide the methodology and interface to collect the data, rules for data acquisition, validation rules etc.

In the next phase the system takes the data collected from the previous phase and performs analysis over it, based on the other inputs like system constraints, applicable rules, domain knowledge and available reasoning algorithms. The analysis engine should be able to decide the best reasoning algorithm to be used based on the contextual information collected from the previous phase.

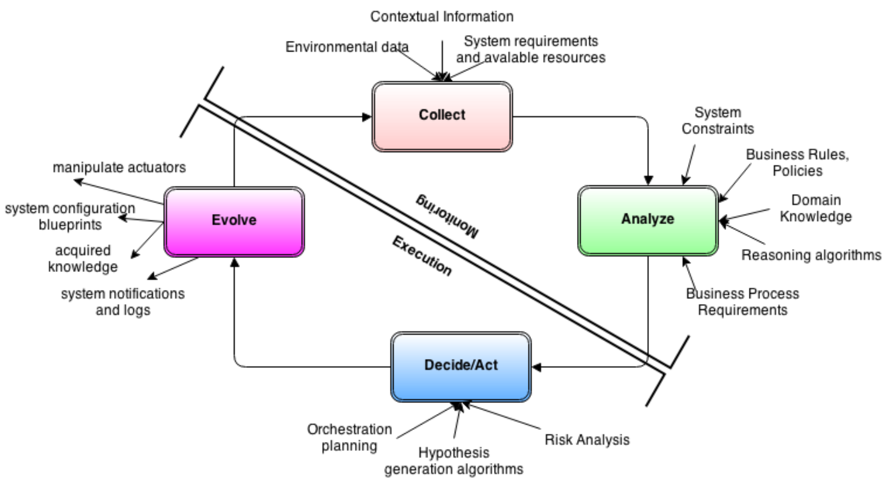

Figure 1 Overall processes of eCPS

Next, the system makes a decision and act accordingly about how to adapt or provide necessary evolutionary in order to fulfill the requirements from the analyzed changes. This phase takes into account of methodology for risk analysis; hypothesis generation based on reasoning over the domain knowledge and provides new orchestration planning with the available services or components. Finally, to decisions or actions formulated have to be implemented in the overall system without distorting the global performance of the system. At this point new system requirements blue prints will be generated for the run-time environment of the system, the acquired knowledge should be fed into the knowledge representation framework and if necessary adjust the actuators for interaction with the physical world. System should maintain proper logs of the changes, any non-strict rules violations and the evolutionary path taken by the system to authorized users.

3.1. Reference Architecture

The eCPS Reference Architecture (eCPS_RA) takes into account of the essential features of the evolvable systems as described in previous sections. The proposed architecture provides the global view of the eCPS with clear layered distinction of various parts. Figure 2 shows the reference architecture, which is divided into five horizontal layers and one vertical layer. Horizontal layers are stacked upward, with the low level computing layers in the bottom and the physical layers in the top most layers. The vertical layer that expands over all the horizontal layers represents the knowledge representation and formal modeling part. The key concept behind this idea is to abstract the computational units from knowledge, models and physical world and provide clear interaction patterns between them.

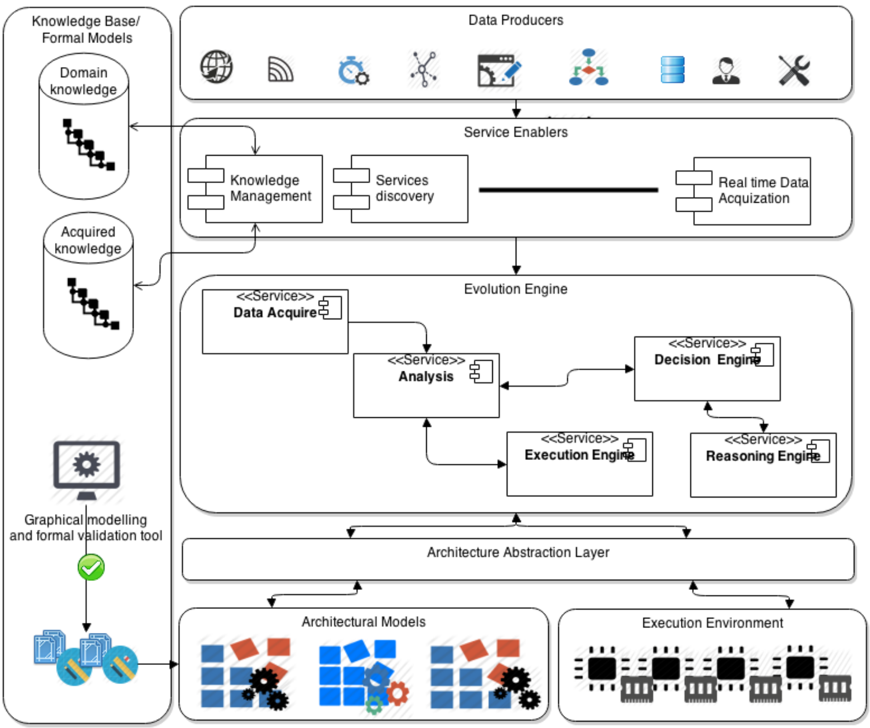

Figure 2 Reference Architecture for eCPS

The knowledge representation layer is used for specifying domain knowledge under which the CPS is functioning. At the same time, this layer also has the mechanism to acquire knowledge at runtime, which will be an important factor for deciding future evolutionary actions.

The system design for complex processes have to be done with the help of graphical tools that allow the simulation and at the same time have the feature of model verifications based on the possible constraints. The outputs of such design procedure are system blueprints, describing the functionalities, requirements and possible interfaces for interaction of each individual component. These blueprints will form the basic building blocks for defining architectural models. The architectural models are thus the collection of individual components (hardware specification and reference implementation, service specification and/or implementations, software components specification and implementations) and orchestration procedures between the components. The architectural models are represented in machine readable standard formats, which will be interpreted by the architecture abstraction layer to find the most suited execution environment for the designed system. Evolution engine comprises of various components that perform tasks as described in the eCPS evolution process in Figure 1. This layer interacts with the architectural abstraction layer for the execution of the evolution strategy. The components of evolution engine make use of services or software components in the service enabler layer to accomplish some particular tasks. Service enabler layers is thus a component engine for deployment of specific services that will help the evolution engine to perform better, like for instance in the reference architecture, we show services for knowledge management, service discovery, data acquisition etc.. The top most layers show the physical world, which is the producer of data. This layer is interfaced with the evolution engine via the service enabler layer, so that the need to integrate new form or data or new protocols for data acquisition can be integrated with the main system, without having to change the overall system.

3.2. Technology Adaptation

The instantiation of the proposed architecture is still in the working phase. This work adopts services and enablers being developed by projects under the European FI-PP program like FI-WARE[1] and FITMAN[2] which provide enablers for data acquisition from the physical world (Internet of Things Service Enablement), complex event processing, context aware data handling, data analysis etc. The outcomes of these projects play an important role for instantiation of the service enabler layer. Various formal knowledge representation standards like OWL, SENSORML etc. are being adopted for the design of the robust knowledge representation framework. In the current state of the research work, the detailed design requirements of the analysis engine and the modeling and validation tool is under study by adopting the principles from Object-Process Methodology (OPM) [16] which provides holistic framework for complex systems modeling, specification, design and verification.

4. Conclusion and Future work

In this paper, we have provided a detailed explanation of the evolutionary cyber-physical systems starting with the necessary characteristics of the system. This research work also involves study on evolutionary systems along with the necessary analysis of the dependent research domains, which have close relation with eCPS. In the last section, reference architecture is proposed that will be used for the realization of eCPS. All the components of the reference architecture are explained with highlights on their

[1] http://www.fi-ware.org/
[2] http://www.fitman-fi.eu/

role, functionality and justification of their need. We have also provided a short discussion on technology adoption to explain the as-is technical state of the research work, towards the realization of eCPS.

The major future work is developing a complete instantiation of the reference architecture and the validation of the proposed architecture based on real scenario. A number of business scenarios mainly in the manufacturing domain can adopt CPS architecture to improve the manufacturing process. Thus, after the development of the system, it will be validated with some manufacturing research partners of our research lab. The major future tasks are thus to analyze the performance of the proposed system in the real scenario. Then based on the observed results, it will be important to improve the system for optimized performance and ease the process for adoption by the manufacturing industry.

Acknowledgments

The research leading to these results has received funding from the EC 7th Framework Programme under grant agreement n° FITMAN 604674 (http://www.fitman-fi.eu) and project SHbuildings SOE3/P1/E508 - programa de Cooperação Territorial Europeia INTERREG IV B SUDOE.

References

[1] M. Onori, A low cost robot programming system for FAA cells, 1997.
[2] M. Onori, D. Semere, and B. Lindberg, Evolvable systems: an approach to self-X production, *Int. J. Computer Integrated Manufacturing*, vol. 24, no. 5, 2011, pp. 506–516.
[3] Z. M. Bi, L. Wang, and S. Y. T. Lang, “Current status of reconfigurable assembly systems,” *Int. J. Manufacturing Research*, vol. 2, no. 3, 2007, p. 303.
[4] T. Ribeiro and G. Gonçalves, “Formal methods for reconfigurable assembly systems,” *Emerging Technologies and Factory Automation (ETFA)*, February, 2010.
[5] K. K. Mittal and P. K. Jain, An Overview of Performance Measures in Reconfigurable Manufacturing System,” *Procedia Engineering*, vol. 69, 2014, pp. 1125–1129.
[6] F. P. Maturana, R. J. Staron, K. H. Hall, P. Tichý, P. Slechta, and V. Marík, An Intelligent Agent Validation Architecture for Distributed Manufacturing Organizations, in *BASYS*, 2004, vol. 159, pp. 81–90.
[7] L. M. Rocha, Selected Self-Organization, *Evolutionary Systems: Biological and Epistemological Perspectives on Selection and Self-Organization*, 1998, pp. 341–358.
[8] D. Semere, J. Barata, and M. Onori, Evolvable Assembly Systems: Developments and Advances, in *Assembly and Manufacturing, 2007. ISAM ’07* , 2007, pp. 282–287.
[9] M. Onori and J. B. Oliveira, Outlook report on the future of European assembly automation, *Assembly Automation*, vol. 30, no. 1, 2010, pp. 7–31.
[10] A. Maffei, M. Onori, P. Neves, and J. Barata, Evolvable Production Systems: Mechatronic Production Equipment with Evolutionary Control, in *DoCEIS*, 2010, vol. 314, 2010, pp. 133–142.
[11] J. H. Holland, *Emergence: From Chaos to Order*, Perseus Publishing, 1999.
[12] K. A. Delic and R. Dum, On the Emerging Future of Complexity Sciences, *Ubiquity*, vol. 2006, no. March, Mar. 2006, pp. 2:1–2:1.
[13] M. Mitchell and M. Newman, “Complex systems theory and evolution,” *Encycl. Evol.*, pp. 1–5, 2002.
[14] P. Trueba, A. Prieto, F. Bellas, P. Caamaño, and R. J. Duro, Self-organization and Specialization in Multiagent Systems Through Open-ended Natural Evolution, in *Proceedings of the 2012T European Conference on Applications of Evolutionary Computation*, 2012, pp. 93–102.
[15] A. F. Smullyan, Morris Charles. *Signs, language, and behavior*. Prentice-Hall, Inc., New York 1946,”J. Symb. Log., vol. 12, no. 02, pp. 49–51, 1947.
[16] D. Dori, *Object-Process Methodology: A Holistic Systems Paradigm; with CD-ROM.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002.