Self-similar Computing Structures for CPSs: A Case Study on POTS Service Process

Dorota Stadnicka[1], Massimiliano Pirani[2(✉)], Andrea Bonci[2],
R.M. Chandima Ratnayake[3], and Sauro Longhi[2]

[1] Faculty of Mechanical Engineering and Aeronautics, Rzeszow University of Technology,
Al. Powstancow Warszawy 12, 35-959 Rzeszow, Poland
`dorota.stadnicka@prz.edu.pl`

[2] Department of Information Engineering DII, Università Politecnica delle Marche, Ancona, Italy
`massimiliano.pirani@gmail.com, {a.bonci,sauro.longhi}@univpm.it`

[3] Faculty of Science and Technology, Department of Mechanical and Structural Engineering
and Materials Science, University of Stavanger, Stavanger, Norway
`chandima.ratnayake@uis.no`

Abstract. This paper proposes a novel method for the structuring of the knowledge of a service process in order to be processed by lightweight declarative computing infrastructures. Through the identification of self-similarities in the process, the flow of the structured information and the sequence of activities performed in the process are easily implemented by means of cyber-physical systems technologies, in order to timely meet the customer/stakeholder's requirements. The study was performed in a telecommunication service providing organization. Service teams create a collaborative network. With the use of the CPS proposed in this work they can communicate problems and disseminate solutions. This methodology uses the information of a set of performance indicators of the service organization to achieve a better control of the effectiveness and the bottlenecks in the supply network. The methodology is borrowed from the mechatronics field and it is prone to a natural extension and reuse for the similar information structures in manufacturing processes.

Keywords: Cyber-physical system · Service process · Business process management systems · Protocols and information communication · Internet of services and service science

1 Introduction

In this work, the authors propose a multidisciplinary link between two research activities. Both are connected with the concept of Industry 4.0. The first one supports horizontal integration presenting a value flow in manufacturing and service processes by Value Stream Mapping (VSM). The other one is a new computing approach for highly dispersed networks of intelligent automation entities that support human tasks and decisions. The new computing trend refers mainly to the so-called Cyber-Physical Systems

(CPSs). The main research question of this work is whether and how the CPS can be implemented to enhance a collaborative network of service teams.

The methodology proposed here puts process experts in contact, using the VSM methodology, with the programming of the process control system through a computing infrastructure based on declarative paradigms – near to natural language. There are two reasons why the authors decided to use VSM. First was that the VSM was already used in the process analysis and wastes identification of the POTS process. Therefore, this tool was already known in the organization. The second reason was that more sophisticated tools such as, for example, BPMN (Business Process Model and Notation) were too difficult to be understood and adopted by people from the analysed service organization, since they are not IT engineers. Therefore, the Value Stream Map (VSMap) became the base of the development of the knowledge representation method and coupled with the database-centric techniques [1, 2], which are extended to cover the multidisciplinary aspects of tiny agents imitating the knowledge of production or services process experts. This multidisciplinary method will follow the state-of-the-art approaches [3, 4] that capture human experts' decisions, data and actions into the structured knowledge in order to render human and machines collaborative agents in the process design and problem solving. The used case will be conducted on an already well-tested and established procedure for plain old telephone services (POTS) [5]. The key of the methodology is in producing a knowledge representation that develops a VSMap into a relational form. That form allows to answer, with a declarative approach, the fundamental questions ('Who', 'Why', 'Where', 'How' and 'What') that capture the human experts' knowledge into a computing infrastructure. The data and knowledge input into the infrastructure determines the evolution of the process and controls its performance, bottlenecks as well as appearing problems. This a novelty in relation to previously published works.

In Sect. 2, some technological background is provided. Section 3 presents the methodology and the main concepts. In Sect. 4, a case study on POTS service is used to materialize the concepts and explain implementation techniques as means for their validation on the experimental set-up. Section 5 provides the conclusion.

2   Background

A rational agent is defined as an actor that should select an action which maximizes its performance measure for each possible percept sequence and evidence provided by agent's built-in knowledge [6]. An artificial agent is composed by an architecture and a program that transform percepts (inputs) into actions (outputs). The control, based on a logical-reactive (condition-action rules) knowledge model, has the advantage of the naturalness of the "if-then" rules and the possibility of representation of declarative and procedural knowledge [7]. The database-centric architecture [2] is a suitable choice in this sense. The development of technical systems and technological processes opens new opportunities both at the level of embedded control systems of a different scale, and at the level of a group interaction of decentralized multi-agent systems [7–9]. Agent-based automation systems, which realize the vision of CPSs, include increasingly

intelligent solutions for managing the information that flows between different layers of products manufacturing or services for the understanding and control of the processes. The information is consumed and generated from the process actors in both human and machine-readable form [10]. The CPS vision implies that the information and the devices will possibly tend to disappear in a dust of collaborating distributed computing elements [11]. This way, the concept of performance measure of the process can be scaled down even to the lowest levels, while all the process elements still remain holistically connected by sharing the same self-similar semantics [1]. The database-centric technology should be based on the capabilities of the full relational model (RM) [12], in a form of a database language and algebra that is expressive enough to encompass both the artificial intelligence aspects and the automated procedures generation. Unfortunately, RM is still not available in lightweight implementation for embedded devices. Nevertheless, the already available database language of SQLite can be applied in dynamical performance optimization problems and smart bottlenecks detection on the highly distributed agent-based systems [1] validated on devices currently priced less than 5$. In this work, the problem of the control of a service process through a typical reflex agent is analyzed.

3   Methodology: Agent-Based Problem Representation and VSM

The methodology suggested in this paper is based on VSM, where the activities can be represented in a descriptive form [13]. In order to be able to use in CPSs, the analysed VSMap [5] is extended in this paper to illustrate additional information (see Fig. 2). The extended version of the VSMap contains the answers for the following questions concerning each step of the process: 'Who', What', 'How', 'Why', and 'Where', as well as the possible challenges/problems that shall be encountered together with countermeasures (see Fig. 2).

The problems in the selected case study process are taken into consideration based on the pre-defined set of rules (i.e. based on experts' knowledge) arranged in a hierarchical pattern. The knowledge gathered from the extended VSMap is used to construct the rule set that controls the evolution and the sequence of the tasks of the selected case study process. Additional information is collected from the analysis of the challenges/ problems, which were discovered in the selected case study process. Therefore, it is possible to present the overall business process in terms of business *STEPS*, which correspond to the tasks to be performed. Hence, in different *STEPS*, it is possible to represent the relevant questions as well as the challenges/problems that are highlighted by experts (i.e. an extension of the VSM approach presented in [5]).

It is possible to find a convenient way to express a business process in the form of a tree structure of self-similar tasks [1]. Hence, the decisions made can change the sequence and the structure of the tasks in order to tackle problems dynamically. The division of systems into sub-systems creates a descent on a tree structure. This enables the adoption of the recursive computing structure [1] where the leaf elements will be atomic tasks of the process. Hence, the semantic used for a productive cell is mapped to an atomic activity at the lowest level of the process activity hierarchy. The *STEPS* will be considered as

particular systems in the overall systems-of-systems control structure. A step is composed of sub-steps until the lowest level is reached (end of recursion), and hence, it collapses to a cell at the lowest level. With the mapping, an immediate application of the performance metrics can be obtained [1, 14]. That can be used to measure at run-time the drift of the current sequence of the tasks from the initial plan (desired state). For each step, a certain amount of time is dedicated. The desirability of a state is captured by the overall throughput effectiveness (OTE) that evaluates any sequence of the states. The OTE is recursively computed from OEE (overall equipment efficiency), and it can be determined by the value of OEE for the lower atomic tasks. This value depends relationally on which meaning (semantics) is assigned to the lower level tasks in the process. The OEE, and then OTE, capture the rate of time, speed and quality in the execution of the tasks [14]. As the current study concerns a 'service process', the overall performance efficiency (OPE) is used instead of the overall equipment efficiency (i.e. the term 'performance' is selected instead of the term 'equipment'). The use of OEE/OTE is not new and widespread, along with its measurement techniques in continuous and discrete processes (see for example [15, 16]). The novelty in the present work is the leveraging of its well-known and simple computing structure to render treatable the complex control of processes where information is heterogeneous and shared collaboratively between human and machines.

By following the original OEE expression in [14] but with extended meaning, we can define:

$$OPE = Aeff \times Peff \times Qeff \tag{1}$$

Note that the meaning of the three factors can assume different semantics depending on the interpretation (in formal logic sense [6]) adopted for them. The only requirement is to be a dimensionless quantity in the interval [0,1]. Typically: *Aeff* - "availability efficiency", captures the deleterious effects due to breakdowns, time delays, setups and adjustments of a process; *Peff* – "performance efficiency", captures a performance loss due to reduced speed, idling and minor stoppages in performing a task; *Qeff* – "quality efficiency", captures the loss due to mistakes or reworks in a task.

The lead time plays a significant role as it is inversely proportional to the OPE. For a deterministic task without problems, like END, we simply put OPE(END) = 1, as the lead time does not suffer any delays. Reversely, if OPE(Tx) = 0, it means that the task Tx never reaches completion, and it is a critical bottleneck where the correction intervention is highly needed. However, the OTE of some task series (sequence) is affected negatively from the passing of time as well as of delays.

The 'service process' is seen as a task environment for the agents, that is described by Performance, Environment, Actuators and Sensors (PEAS) [6] (Fig. 1).

In order to gather the knowledge concerning the selected service process, the following information is considered according to the proposed methodology: What are the steps of the process? Who should realize a particular step? What exactly should be done? How to do it? Why should these actions be undertaken? Where should the work be done? What documents will be used (If any)? What documents will be created (If any)? Which data bases should be available for the employees performing the step of the process? What problems can appear in the step? What should be done in case of problems appearance? The questions gather the expert's knowledge and they are
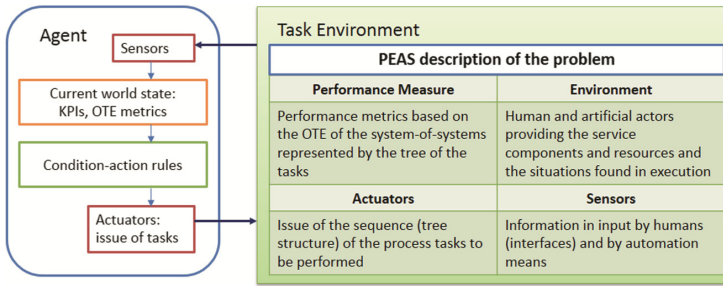
Fig. 1. Representation of the control of the service process through reflex agent scheme.

attributes of the systems and sub-systems (and so steps) with a hierarchical self-repeated structure (a pattern inside of a pattern for the fractal paradigm). The first statement is as follows: The agents participating in the process (machine or human) have a (quick) access to a distributed knowledge base (KB) containing data, documents, semantics, ontologies (if needed), rules and procedures. The information is expressed and modelled in a relational form, following the relational model. The other statement is: The agents can exchange the information with a suitable networking technology and the machine-to-machine or peer-to-peer connections are granted in this infrastructure.

4   A Case Study – POTS Service Process

The case study POTS service process [5] was analyzed using the proposed methodology. The POTS installation process concerns installation of plain old telephone services. Shortly the process can be described as follow. The first step of the process is transmission of an installation order by Polish Telecommunications to a firm providing telecommunication services with all data necessary to perform the installation process. Then a Technical Teams (TT) manager transmits the order to a corresponding team. The installers, who obtained the order check the materials availability on the vehicle and if it is necessary drive to a warehouse for the materials. When the materials are loaded on the vehicle the team go to an installation place. If the client is present they perform the installation and prepare all necessary documentation. In other case they inform TT manager and wait for a decision.

The extended VSMap of the process was developed to illustrate the case study service process together with the experts' knowledge based rules. A fragment of the process representation is shown in Fig. 2. In STEP 1, the question "What if a team is not available?" is going to be posed and verified in relation to the 'Who' attribute. In STEP 2, "What if not all materials are available?" is related to the 'What' attribute: "Checking the status of materials on the car". Typically, a possible problem occurring during the execution of the step may be associated with each of the step attributes (i.e. one of the 5 questions related to the knowledge representation). The answer to the problem creates a specific instance of the possible conditional alternative systems-of-systems structures. It affects the sequence of the subsequent steps. In other words, the instant structure is an instance of all the possible available tree-structures foreseen by the process experts

in order to solve the problems, and conditioned by the evidence of a problem status. This is a novelty with respect to [1] where the structure was considered to be statically determined, though arbitrary. The service process execution adapts itself dynamically to the problems occurrence.
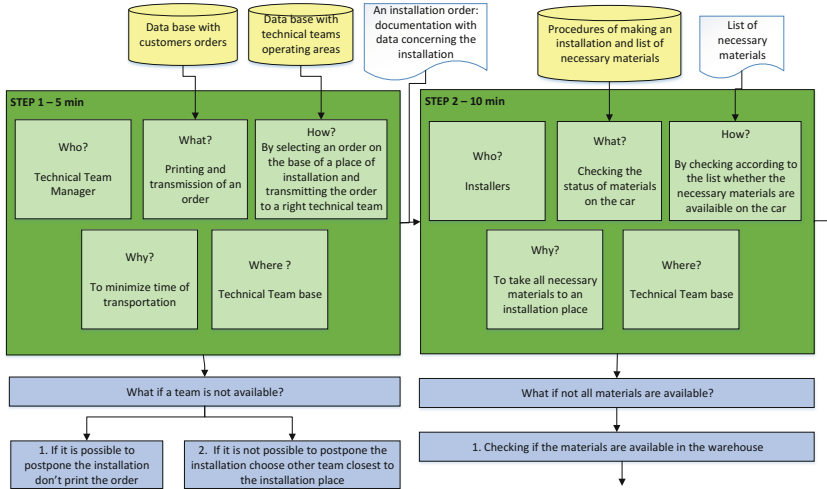

Fig. 2. Graphical representation of the POTS service process – a fragment.

It opens great new possibilities for the presented tool that can be also used in autonomous, evolving and self-organizing systems. The occurring problem appears as a percept to an agent. The agent queries the 5 questions to the knowledge base (or input database) relative to the associated sub-system under control, when the OTE of this subsystem is under the expected quality threshold or is diminished of a certain target rate. Thus, the result is relationally equivalent to the percept information. Depending on the percept, the sub-structure of the current step is changed accordingly.

The whole procedure can be put in a relational form and ready for implementation through the database tables (relations, in general) expressed in Fig. 3. In particular, the core table for the structure instance is the *Condition-action* relation in Fig. 3. This is the plug for artificial intelligence and learning. It can be expressed in other forms as well in order to admit different learning approaches. In the present case, the experience relation is the implementation of a typical decision-tree that captures some of the problem solving skills of human experts. The *Condition-action* table is quite compact as when a NULL (no value) in the cell is present, it means that the perception value can be whatever (undefined). The first column in the *Condition-action* is the identifier of the systems *Sx,x,x*. The comma- separated numbers in this notation indicate the path along the systems structure tree, one index for each of the level descent, in depth dimension, and ordered in breadth along the same level. For example, the S1, S2 are two systems at the top level containing STEP 1 and STEP 2 respectively. S1,1 and S1,2 are at the second level and they appear only when the percept of the problem p1 gives a true output (T). In another case (F), no problems occurred and the structure instantiation process

(decision tree descent) stops on S1, and the task of step1 is executed. The action to be done and the descent in the tree are governed by the Sub_system relation attribute of the *Condition-action* relation (Fig. 3). There are commands to descent to the next level and possibly to establish a structure of descendants, as in the "Next_Level (series)" in the third row.

Conditions				Condition-action (rules or productions)						
Percept	Value	System with OTE under threshold	Semantic	System	Question	p1	p2	p3	...	Sub_system
p1	T	S1	Team not available	S1	Who	T				Next_Level
p2	T	S1,1/S1,2	It is possible to postpone	S1	Who	F				Task(print order)
				S1,1	What		T			Next_Level (series)
p3	T	S1,2,1/S1,2,2	Another team available and near to the site	S1,2	What		F			Next_Level (series)
				S1,1,1						Task(postpone)
p4	F	S2	All the required materials available in the car	S1,1,2						S1
				S1,2,1	What			T		Next_Level (series)
p5	F	S2,1	All the required materials available in the warehouse	S1,2,1	What			F		Task (END)
				S1,2,1,1						Task(choose other team)
...	....	....	....	S1,2,1,2						S1

Fig. 3. Relations that implement the actions through the issuing of questions on the problems detected by the lowering of OTE values.

Other commands tell to simply start executing and controlling the tasks, as when "Task (print order)" or "Task(END)" appears. Otherwise, the substitution of a system with another one, as in the 6th row where S1,1,2 is transformed in S1, can also appear. This creates loops when needed. Given the *Condition-action* relation (here given as a single table for compactness of the paper, but such relation can be in a more structured form across more tables), the agent assigned to a certain step can query about the status of the problem in order to determine the instant system structure as well as the tasks sequence to be performed (actions) in the next business structure instance. For example, the agent produces the following (in SQL pseudocode):

```
SELECT Sub_system FROM experience WHERE
    Question==Who AND p1==(SELECT Value FROM percepts
            WHERE Percept==p1) WHERE System==S1;
```

The result of *SELECT* will command the agent to descend to the next level, and thus, to query again for the systems descendants in the three, i.e. *S1,1* and S1,2, and continue with the queries until it results in a task execution. This recursive procedure can be managed with a single query if the SQL dialect has recursive capabilities, or in the relational model. The result is the compilation of a *systree* table, which enables the computing of the OTE [1]. However, the computing details were not presented here. As an example, the systems generated for STEP 1 are presented in Fig. 4.
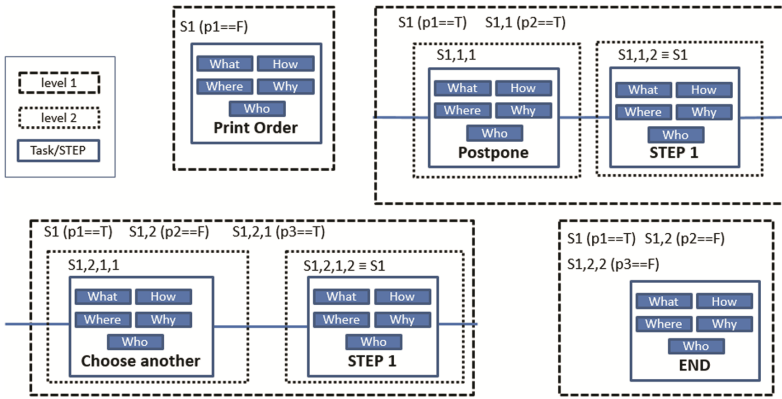
Fig. 4. Graphical representation of the POTS service based on experience table conditions.

The representation shows the structure instantiation obtained with the problem perception from the agents. After an agent obtains the information structure from the *Condition-action* relation, it sees the actual structure and executes it while measuring its OTE. In Fig. 4, all the conditional cases provided in the *Conditions* relation of Fig. 4 are depicted. Each of the four systems in Fig. 4 are obtained through a decision tree. Each of the four systems are alternative structure instances of the process STEP 1, depending on an agent decisions and triggered by the percepts (problems' status) sampled at the beginning of the step. It should be noted that in the presented example the subsystems were considered to have a series structure, but it is not always the case in general [1, 14].

5   Conclusions

This multidisciplinary research work aims at establishing a methodology that links approaches of the process control to the CPSs. From the computer science side, it relies on an extension of a well-known recursive performance metrics method for industrial processes and manufacturing. The tree structure of this computing involves also all the topologies that can be put in a direct acyclic graph form through the four fundamental structures (series, parallel, expansion and assembly) that are proven for completeness in manufacturing layouts [14]. If applied to self-similar structures the computing results greatly simplified and can scale well on tiny devices, also for tree structures of relevant depth. The property of self-similarity is a common feature in manufacturing and other processes, which mostly depends on the appropriate semantics chosen for the indicators. The presented methodology can be straightforwardly applied to the class of problems encompassed from the fractal factory paradigm [1]. The knowledge of experts is conveyed in a declarative (natural-like) language near to humans and readable from machines, in order to program a computing infrastructure for the real-time process control and bottleneck detection. The presented methodology adds value to the well-known VSM method. It was expressed and challenged through a relevant case of POTS

services. The performed analysis allows to conclude that the presented methodology enables its executors to understand the realized process and the problems that may appear better. However, it should be underlined that the process has to be well analysed and structured by humans. They should create the knowledge based on a process analysis as well as people's experience to create a graphical representation of the process and the rules being applied in the process realization. Additionally, possible problems should be discussed in order to find solutions that then will be proposed by the created CPS following the process. The analysed POTS process is ready for the future experimental sessions. In the future works, the proposed performance indicators such as OPE and OTE will be calculated on the basis of the data derived from the POTS service process realization.

References

1. Pirani, M., Bonci, A., Longhi, S.: A scalable production efficiency tool for the robotic cloud in the fractal factory. In: IECON 2016-42nd Annual Conference of the IEEE, pp. 6847–6852 (2016)
2. Bonci, A., Pirani, M., Longhi, S.: A database-centric approach for the modeling, simulation and control of cyber-physical systems in the factory of the future. In: 8th IFAC Conference on Manufacturing Modelling, MIM 2016, pp. 249–254. IFAC-PapersOnLine (2016)
3. Qin, H., Hongwei, W., Johnson, A.L.: A RFBSE model for capturing engineers' useful knowledge and experience during the design process. Robot. Comput. Integr. Manufact. **44**, 30–43 (2017)
4. Bruno, G., Taurino, T., Villa, A.: An approach to support SMEs in manufacturing knowledge organization. J. Intell. Manufact. **23**, 1–14 (2016)
5. Stadnicka, D., Ratnayake, R.M.C.: Minimization of service disturbance: VSM based case study in telecommunication industry. In: 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016, pp. 255–260. IFAC-PapersOnLine (2016)
6. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Egnlewood Cliffs (2009)
7. Vassilyev, S.N., Kelina, A.Y., Kudinov, Y.I., Pashchenko, F.F.: Intelligent control systems. Procedia Comput. Sci. **103**, 623–628 (2017)
8. Leitão, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., Colombo, A.W.: Smart agents in industrial cyber-physical systems. Proc. IEEE **104**(5), 1086–1101 (2016)
9. Zhenqiang, B., Weiye, W., Peng, W., Pan, Q.: Research on production scheduling system with bottleneck based on multi-agent. Phys. Procedia **24**, 1903–1909 (2012)
10. Frank, J.A., Kapila, V.: Integrating smart mobile devices for immersive interaction and control of physical systems: a cyber-physical approach. In: Zhang, D., Wei, B. (eds.) Advanced Mechatronics and MEMS Devices II. MN, pp. 73–93. Springer, Cham (2017). doi: 10.1007/978-3-319-32180-6_5
11. Chiang, M., Zhang, T.: Fog and IoT: an overview of research opportunities. IEEE Int. Things J. **3**(6), 854–864 (2016)
12. Date, C., Darwen, H., Lorentzos, N.: Time and Relational Theory, Second Edition: Temporal Databases in the Relational Model and SQL. Morgan Kaufmann, San Francisco (2014)
13. Stadnicka, D., Ratnayake, R.M.C.: Simple approach for Value Stream Mapping for business process analysis. In: Proceedings of the 2015 IEEE IEEM - IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 88–94 (2015)

14. Muthiah, K.M.N., Huang, S.H.: Overall throughput effectiveness (OTE) metric for factory-level performance monitoring and bottleneck detection. Int. J. Prod. Res. **45**(20), 4753–4769 (2007)
15. Nayak, D.M., Vijaya Kumar, M.N., Naidu, G.S., Shankar, V.: Evaluation of OEE in a continuous process industry on an insulation line in a cable manufacturing unit. Int. J. Innovative Res. Sci. Eng. Technol. **2**(5), 1629–1634 (2013)
16. Muthiah, K.M., Huang, S.H.: A review of literature on manufacturing systems productivity measurement and improvement. Int. J. Ind. Syst. Eng. **1**(4), 461–484 (2006)