Holonic-based Task Scheduling in Smart Manufacturing Systems

Valentin Vlad
Electrical Engineering and Computer Science Department, University of Suceava, Suceava, Romania

Keywords: Smart Factory, Industry 4.0, Holonic Control.

Abstract: The industrial domain undergoes a deep transformation, referred by the technical literature as the fourth industrial revolution. The key element in this transformation is the integration of advanced digital technologies in production, in order to improve the autonomy and interoperability of the participating entities. In order to have a standard-based integration, a reference architectural model was proposed, RAMI 4.0, to guide the migration of the actual production systems to the next generation ones. In this paper we discuss holonic-based solution for dynamical distribution of tasks in a smart manufacturing system, according to the recommendations of RAMI 4.0.

1 INTRODUCTION

The industrial domain undergoes a deep transformation, referred by the technical literature as the fourth industrial revolution (Industry 4.0). The key element in this transformation is the integration of advanced digital technologies in production, in order to improve the autonomy and interoperability of the participating entities throughout the life cycle of products.

An important component of I4.0 concept is represented by the Smart Factories in which humans, machines and resources communicate with each other, like within a social network. In this aim, the production devices are supposed to include intelligent software components, enabling them to autonomously control the execution of their task and cooperate with each other for achieving the global goals of the system they are part of. The communication between these Cyber-Physical Systems (CPS) is based on Internet of Things (IoT) and Internet of Services (IoS) technologies, implying the use of a service-oriented architecture (SOA) in which each element of the value chain can be accessed as services from other elements (Contreras et al., 2017).

To have a structured and standard-based integration of these technologies, the promoters of the Industry 4.0 concept developed a set of approaching guidelines in form of an architectural model named RAMI 4.0 (Reference Architectural Model for Industry 4.0). RAMI 4.0 describes also the properties that CPS must meet in Industry 4.0. They are seen as I4.0 components with the cyber part represented by an "administration shell", designed to provide a description of the physical part in the information world. The administration shells include a series of 'sub models', which represent different aspects of the physical devices. These 'sub models' are to be standardized so as a specific machine can be easily found among many others I4.0 components. Several I4.0 components can be grouped into a composite component and exhibit aggregated functionalities through a high-level administration shell, in the same way as individual components (Liu and Xu, 2017).

These concepts developed in RAMI 4.0 make the agent paradigm a very good candidate for developing the smart factory goal of Industry 4.0 (Adeyeri et al., 2015, Lu, 2017). Moreover, the holonic concepts capture very well the properties of I4.0 components, namely the autonomy, cooperation and recursive encapsulation. In this paper we discuss a holonic-based solution for dynamical distribution of tasks in a smart manufacturing system, according to the recommendations of RAMI 4.0.

2 HOLONIC-BASED STRATEGIES FOR TASKS SCHEDULING

Within a holonic system the scheduling of tasks can be realized in a dynamical way, according to the

Vlad, V.
Holonic-based Task Scheduling in Smart Manufacturing Systems.
DOI: 10.5220/0007900202420245
In *Proceedings of the 8th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2019)*, pages 242-245
ISBN: 978-989-758-373-5

status and loading of holons, through negotiation activities. This work investigated two strategies for task scheduling.

The first strategy is to conduct direct negotiations between the order holons that coordinate the execution of products and the holonic devices in the system that perform their actual processing (Figure 1). Considering the case of a simple product, the corespondent order holon will negotiate with all devices in the system capable of executing the product operations and, based on the received offers, will establish an execution plan that will minimize the time to complete the product. Problems in the operation of production equipment, or changes in planning due to, for example, urgent orders, will be announced to the order holons, who will be able to re-plan their activities. Similarly, the introduction of new equipment into the system can be announced to the order holons to perform replanning.

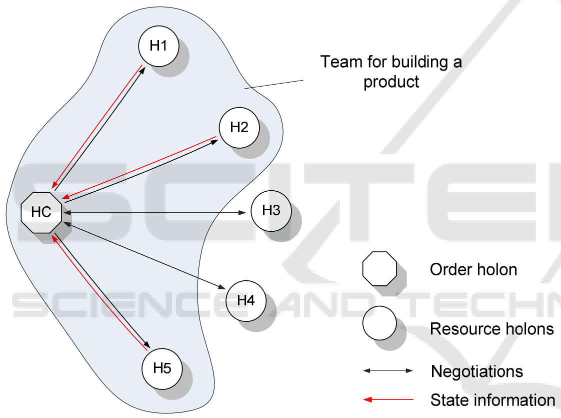

Figure1: Task scheduling through direct negotiations between order holons and resource holons.

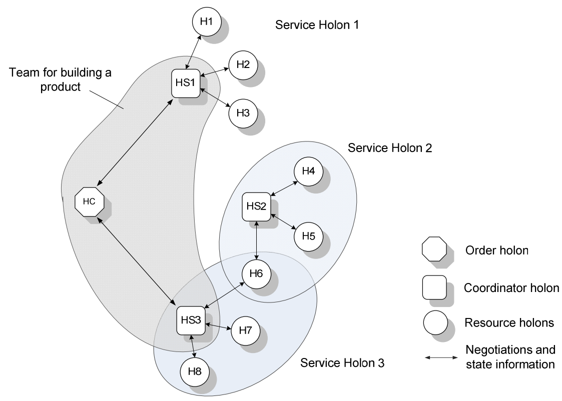

Figure 2: Task scheduling through hierarchical negotiations.

The second, more advanced strategy, is to organize the holonic devices in the system in holons with complex intelligence (*service holons*), depending on the services they provide. In this case, for a particular task, an order holon will no longer negotiate with each holon device that has the service involved in the task, but only with the coordinator of the complex holon corresponding to the service, as illustrated in Figure 2. The coordinating holon will negotiate with the subordinated holonic devices in order to determine an optimal planning solution that will be then transmitted to the order holon.

Within this organization, coordinator holons can optimize system performance by looking for a balanced load of subordinate holons. This balancing can be provided for a certain time horizon, called *the optimization horizon*, with a duration equal to a fraction of the duration of the considered working session. Considering for a holonic device the notations for tasks and temporal constraints illustrated in Figure 3, it is possible to define the load of the holon at discrete moment k, I_k, of the form:

$$L_k = \frac{1}{\mu_k} \cdot \frac{D_p^k}{D_o^k} \tag{1}$$

where:

- D_p^k – the total duration of tasks in the holon's agenda (planned for execution or in progress) at discrete time *k*;
- D_o^k – duration of optimization horizon at discrete time *k*;
- μ_k – average task processing rate at discrete time *k*.

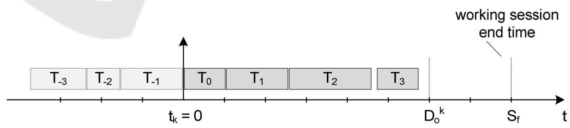

Figure 3: Notation example for task and temporal constraints of a holonic device

The planned duration of tasks in a holon's agenda, at discrete time k, can be expressed by the relationship:

$$D_p^k = \sum_{i=0}^{n} d_p(T_i^k) \tag{2}$$

where:

- $d_p(T_i^k)$ – the planned duration for the i-task at discrete time *k*;
- n – total number of tasks in holon's agenda.

The optimization horizon at time k will be given by the relation:

$$D_o^k = \begin{cases} D_o, & D_o \le S_f - t_k \\ S_f - t_k, & D_o > S_f - t_k \end{cases} \quad (3)$$

where:

- D_o – a predefined time for the optimization horizon;
- S_f – session completion time;
- t_k – the current time.

The average task processing rate at discrete time *k*, μ_k, can be defined as the arithmetic mean of the processing rates of an arbitrary number of recently completed tasks and the processing rate of the task being executed, according to relationship (4). Within this relationship, a task's processing rate is defined as the ratio between the scheduled duration d_p and the actual processing time d_r of the task.

$$\mu_k = \frac{\sum_{i=-1}^{-m} \frac{d_p(T_i^k)}{d_r(T_i^k)} + \varepsilon_k \frac{d_p(T_0^k)}{d_r(T_0^k)}}{m + \varepsilon_k} \quad (4)$$

where:

- $d_r(T_i^k)$ – the actual processing time of the task with index *i* at discrete time *k*. Negative value of the index has the meaning of *completed task*.
- T_0^k – the task being executed at discrete time *k*;
- m – the number of the most recent completed tasks considered for determining the task processing rate
- ε_k – validating coefficient.

The processing rate of the current task has significance within equation (4) only if it causes a deterioration of the average processing rate, allowing to reflect the current holon problems in the value of its load. Validation or invalidation of this term is achieved by the coefficient ε_k, defined as follows:

$$\varepsilon_k = \begin{cases} 0, & \frac{d_p(T_0^k)}{d_r(T_0^k)} \ge \mu_{k-1} \\ 1, & \frac{d_p(T_0^k)}{d_r(T_0^k)} < \mu_{k-1} \end{cases} \quad (5)$$

where:

- μ_{k-1} – mean processing ratio of tasks at discrete moment *k-1*.

Therefore, the occurrence of delays in completing the tasks (due to defects or delays in the delivery of the semi-finished products) will lead to a decrease in the task processing rate and implicitly an increase in the loading of the holons.

According to the relation (1), the loading of a holon at discrete time *k* can have a value:

- *sub-unitary*, that is, $L_k \in [0, 1)$, in which case the holon is considered *under loaded*;
- *equal to the unit*, $L_k = 1$, corresponding to a 100% loading of holon;
- *higher than one*, $L_k \in (1, \infty)$, in which case the holon is considered *overloaded*.

The overload of a holonic device h_i, at discrete time k, can be defined by the relationship:

$$S_k(h_i) = L_k(h_i) - 1 \quad (6)$$

and may have positive values (overloaded holon), negative (underloaded holon), or may be zero (100% loaded holon).

It is considered that a holon can enter into an alert state when its overload at discrete time *k* exceeds a certain threshold, called *alert threshold*. The value of this threshold at time *k*, T_{av}^k, can be given by the relation:

$$T_{al}^k = \begin{cases} T_{al}, & D_o < S_f - t_k \\ 0, & D_o \ge S_f - t_k \end{cases} \quad (7)$$

where:

- T_{al} –a predefined threshold value that can be identical for all holonic devices

According to the relationship (7), when the remaining time until the end of the current session becomes less than the predetermined optimization horizon, the alert threshold value becomes 0 so that any overload of the holon will lead to an alert state.

Considering a holon with complex intelligence H_i containing several holonic devices *hi*, $H_i = \{h_1, h_2, \dots h_n\}$, we can define the maximum and minimum overloads of holon H_i at time k, as follows:

$$S_{max}^k(H_i) = max\{S_k(h_1),\ S_k(h_2), \dots S_k(h_n)\} \quad (8)$$

$$S_{min}^k(H_i) = min\{S_k(h_1),\ S_k(h_2), \dots S_k(h_n)\} \quad (9)$$

A complex holon is considered to be in a state of emergency when at least one holonic device in its composition is in an alert state, that is $S_{max}^k(H_i) > T_{al}^k$. A complex holon in a state of emergency will no longer enter into negotiations for accepting new tasks, but will try to redistribute the tasks among the holons of its holarchy in order to eliminate all the alert states. Negotiations can be resumed once the alert states are eliminated, or when it is no longer possible to transfer tasks among holons for reducing their overloading.

A holonic device h_i can accept new tasks, or transfered from other holons, provided they do not lead to an overload, according to the relationship:

$$S_k(h_i) + \frac{d_p(T_x)}{D_o^k} \leq 0 \quad (10)$$

or

$$I_k(h_i) + \frac{d_p(T_x)}{D_o^k} - 1 \leq 0 \quad (11)$$

Transfer of tasks within a complex hollow Hi can be initiated both if a holonic device in its composition enters an alert state (relationship 12), and when the difference between the maximum and minimum holon overload exceeds a threshold, called *transfer threshold* T_{tr} (relationship 13).

$$S_k(h_i) > T_{al}^k,\ h_i \in H_i \quad (12)$$

$$S_{max}^k(H_i) - S_{min}^k(H_i) > T_{tr} \quad (13)$$

This solution allows a continuous adaptation of the system in presence of perturbations and an optimization of production through a balanced distribution of tasks between production facilities.

3 CONCLUSION

In conclusion, the application of holonic concepts in the field of manufacturing systems allows the development of dynamic and interactive control solutions, with the potential to ensure both a rapid response of the system to changes and an efficient use of its resources. The industrial acceptance of these solutions, however, continues to require significant effort in the development of architectural models, implementation platforms and case studies to ensure the effectiveness of holonic industrial control, both technically and economically.

This paper presented two holonic task scheduling solutions for intelligent manufacturing systems. The first solution, characterized by a flat organization of resource holons, is efficient in dealing with perturbations generated by workstation failures, but implies a high complexity when a continuous balanced distribution of tasks is desired.

The second solution considers a holarchical organization of resource holons. Compared with the first one this approach exhibits a higher adaptability both in dealing with perturbations due to workstations malfunctions as well as in redistribution of tasks when a new device is added to the system.

REFERENCES

Contreras, J.D., Garcia, J.I., Pastrana, D, Developing of Industry 4.0 Applications, *iJOE*, Vol. 13(10), 2017, pp. 30-47.

Lu, Y., Industry 4.0: A survey on technologies, applications and open research issues, *Journal of Industrial Information Integration*, Vol. 6, 2017, pp. 1-10.

Liu, C., Xu, X. Cyber-physical Machine Tool – The Era of Machine Tool 4.0, *Procedia CIRP*, Vol. 63, 2017, pp. 70-75.

Adeyeri, M.K., Mpofu, K., Olukorede, T.A. Integration of agent technology into manufacturing enterprise: a review and platform for industry 4.0, *2015 International Conference on Industrial Engineering and Operations Management (IEOM), IEEE (2015)*, pp. 1-10.