HAL open science

Modelling digital twins as a recursive multi-agent architecture: application to energy management of communicating materials

Hang Wan, Michael David, William Derigent

To cite this version:

Hang Wan, Michael David, William Derigent. Modelling digital twins as a recursive multi-agent architecture: application to energy management of communicating materials. 17th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2021, Jun 2021, Budapest (virtual), Hungary. hal-03276817

HAL Id: hal-03276817

https://hal.science/hal-03276817v1

Submitted on 2 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modelling Digital Twins as a recursive Multi-Agent Architecture: application to energy management of communicating materials

Hang WAN*, Michael DAVID*, William DERIGENT*

* Université de Lorraine, CRAN, UMR7039, Campus Sciences, BP70239, 54506, Vandœuvre-les-Nancy Cedex, France
(e-mail: firstname.lastname@univ-lorraine.fr)

Abstract: In recent years, there has been an increasing interest in intelligent product especially for industrial applications. A special intelligent product was proposed ten years ago, and this paradigm was named "communicating material", i.e. materials equipped with micro components able to sense, store and communicate data. Communicating material needs a recursive architecture to be managed with a digital twin. A literature analysis is proposed concerning recursive multi-agent systems and architectures regarding communicating material applicability. The most promising reference architecture was adapted for modelling digital twin of a communicating material under energy constraint. The composition and decomposition mechanisms are detailed to improve energy performance by generating an aggregated product with a global view. An applicative scenario relative to the different aggregation modes between two communicating materials, illustrates the proposed architecture ability.

Keywords: Communicating materials, digital twin, recursive multi-agent system, architecture modelling

1. INTRODUCTION

The rapid development of Internet of Things (IoT) leads to the intelligent product paradigm (Wong et al., 2002). In this vision, manufactured products can complete tasks independently and can also cooperate with other products to respond to changes in environment. This adaptive ability makes it widely used in industry (from manufacturing to logistics) (Meyer et al., 2009).

In 2009, the "communicating material" (CM) concept was proposed by (Kubler et al., 2010). It is a special type of intelligent product. This material can communicate with environment. More specially, they can process, store monitoring data in their own structure and exchange information with other CMs or users. Compared with traditional intelligent product, CM has two more special abilities: the capability of being intrinsically and wholly communicating which support physical transformation (decomposition/composition); and the capability of managing its own data. This concept has been applied in different works. A recent example is the communicating concrete in the McBIM project (Derigent et al., 2019). A Wireless Sensor Network (WSN) is embedded into a concrete component to monitor, process and store structural health information for construction industry. CM works in a resource constrained environment which brings challenges in data and energy management (Wan et al., 2020). Indeed, in this project, the communicating concrete should work from manufacturing to exploitation (at least ten years).

To meet this long-term service requirement, solutions are required to monitor and forecast remaining energy of the WSN embedded in the materials. Distributed solutions over the network are often suboptimal. Using a virtual representation could allow to achieve a global optimization. Indeed, applying the concept of Digital Twin (DT) seems to be a suitable solution, since it would bring a WSN virtual model to store all information concerning the physical part and provide optimal behavior decisions (with a centralized solution) to extend its lifetime. DT concept was first introduced by Michael Grieves in 2002 and formalized in 2016: a digital twin is a set of virtual information constructs that describes the physical product. It includes all information that could be obtained from the physical part (Grieves and Vickers, 2016). Many other DT definitions have been proposed. Form Rosen, a DT is realistic models of the current state of process and behaviors which can interact with real environment (Rosen et al., 2015). (Gabor et al., 2016) define a DT as the simulation of physical object to predict future state of the system. A comparison of these different views is provided by Negri (Negri et al., 2017). In our definition, a DT is **a digital representation of real CM products, which uses the input data to mirror, predict and provides energy efficient solutions to extend CM lifetime**.

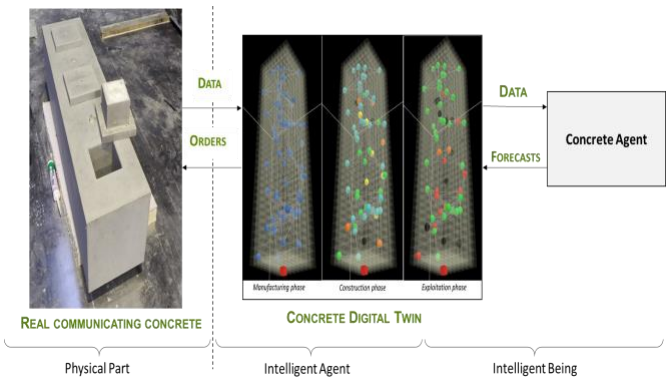

Fig. 1. Digital Twin for communicating concrete

The DT example for communicating concrete is shown in Fig. 1. Real communicating nodes (CNs) are represented by agents in concrete DT. Colors of agent represent energy level of CN from 100% (blue) to 0 (black). Node agents are grouped in a concrete agent which can store all product information and monitor data of the real concrete. Concrete agent can provide optimal orders with a global view to meet environment changes. These orders could be a monitoring frequency change, a new aggregation technique application at node or a different routing strategy for data collection.

During all lifecycle phases, each concrete element can manage its own data and make own decision to accommodate environmental changes. Besides, it can aggregate with other elements into an aggregated product with comprehensive knowledge of energy information. This flexible aggregation ability brings challenges to their dedicated DTs: How to construct a DT with many agents (multi-agent) and with many abstraction levels? How to realize the recursion that aggregate multiple independent DTs into an "aggregated DT" to improve performance and extend lifetime?

To address the mentioned issues, this paper aims to design a recursive multi-agent architecture to support this kind of management. In next section, the recent recursive holarchy and multi-agent architecture are reviewed. The CM recursive agent model is detailed in section 3. A global view of the recursive architecture and the composition/decomposition mechanisms are presented in section 4. CM application scenario is given in section 5. Finally, section 6 concludes this paper and presents some perspectives for future work.

2. RELATED WORK

The concept of intelligent product has been widely applied in our daily life. The recursion of multi products has attracted a lot of attention, especially for the design of control system.

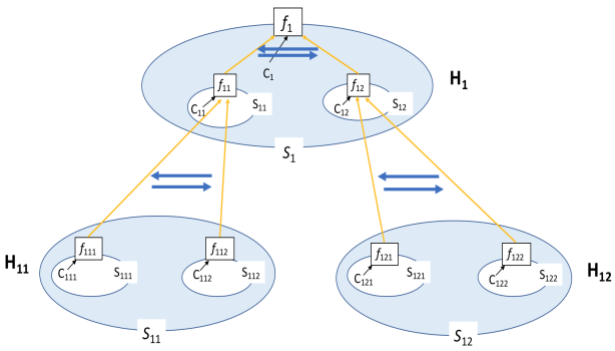

Fig. 2. Holonic architectural model for embedded diagnosis

(Le Mortellec et al., 2013) present SURFER, an embedded holonic fault diagnosis architecture for complex transportation systems (CTS) where train, vehicles and doors are considered as intelligent products. The generic holonic architecture is illustrated in Fig. 2. All elements are represented by Holons. The term Holon was proposed by (Koestler, 1968). A holon consists of an information processing part combined with a physical part. In this architecture, H_1 represents a train. It has two children (vehicles H_{11} and H_{12}). Besides, each vehicle also has two doors. The system S_1 is decomposed into subsystem S_{11} and S_{12}, S_{112} is a sub-system of S_{11}. In this hierarchical structure, each level takes results from lower layer into account and report results to upper layer. This approach presents information exchanges on the same level and the recursion for different levels. The authors describe a modeling framework but does not detail the composition and decomposition mechanisms needed to manage a dynamic system.

(Sallez et al., 2015) introduce a recursive holarchy in logistics for Physical-Internet or (PI)-containers. These containers have some communicational and decisional capabilities. According to their size and function, three types are defined from the largest to the smallest: transport container, handling container and packaging container. Encapsulation and composition concepts are defined for the composition of containers. Indeed, containers of the same type can be composed as a "composite" PI-container. Small containers (handling and packaging) can be encapsulated in a bigger one. However, the compositional mechanisms are not formally specified.

(Esmaeili et al., 2017) use a location information-based algorithm to construct the holonic multi-agent system (HMAS) for large scaled applications (LSA). In the proposed four-level architecture, a holon is selected as the head at each level except for the fourth layer. Holons can communicate with those at same level, or with the holons in higher or lower levels through the heads. This location information-based system can work well with static applications. However, the energy cost for reconstruction with this model is very high, it is hard to meet the high flexibility requirement of CM application.

A recursive agent meta model (RAMM) was proposed by (Vo et al., 2012) for multi-scale modeling (MSM). This meta-model has four main parts: species, agent, environment, and scheduler. Agents of different species can form a hierarchical structure. For example, a root agent located at level *0* consists of two agent species at level *1* (species *1* and species *2*). At the lower level, species *3* is a micro species of species *1*. The authors define the capture and release methods to add or remove micro species. However, the interactions between agents are not detailed.

(Occello et al., 2019) present a generic Recursive Multi-Agent System (MAS-R) for supervision and observation of large-scale complex systems. A MAS-R agent has four parts: Knowledge (K), Observation (O), Composition (C) and Recursive Interaction (RI). Basic knowledge and abstraction level of agent are stored in K part. O is responsible for observing the state of composition. Composition tasks are performed in C. The RI part of an agent updates its own data, processes, and transfers data with other agents.

The hierarchical MAS-R structure is shown in Fig. 3. At the level *0*, agents a_0^i are "Elementary" agents which have a physical applicative part and a recursive agent part. The agent a_i^j represents the agent *j* on level *i*. These agents on high levels (*i* is greater than zero) are called "Partial" agent (no physical part). A "Complete Elementary" agent is composed by its elementary agent and all theirs directly related "Partial" agents (from abstraction levels). "Composed" agents are the gathering of few "Partial" agents from the same abstraction level. Besides, the authors also detailed the composition mechanism. Once composition conditions of an agent are met,

it creates parent agent and invites others to join. This agent then adds accepted agents to the aggregated agent list. At the end, it sends "update" messages to all agents in the list to ensure that they have the same parent.

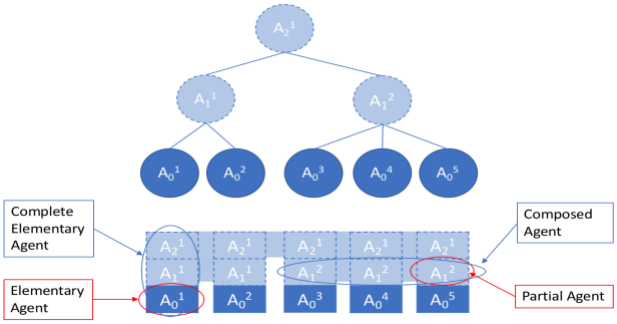

Fig. 3. The structure of MAS-R

The concept of recursion has been discussed or applied in all mentioned works. A summary is given in Table 1. We try to apply a dynamic recursive architecture to CM applications. Four characteristics are used to analyze these presented works: application domain, architecture, formalization of agents and aggregation mechanism. Application domain presents their applied filed. The architecture records whether the approach is fixed or dynamic. Whether they proposed an agent model or composition mechanisms are recorded in formalization of agent and aggregation mechanisms respectively.

Table 1. Summary table of recursive MAS

Reference	Application domain	Architecture	Formalization of agent	Aggregation mechanisms
SURFER (Mortellec et al., 2013)	CTS	Fixed	No	No
PI-Container (Sallez et al., 2015)	Logistics	Dynamic	No	No
HMAS (Esmaeili et al., 2017)	LSA	Fixed	No	Yes
RAMM (Vo et al., 2012)	MSM	Dynamic	Yes	Yes
MAS-R (Occello et al., 2019)	WSN	Dynamic	Yes	Yes

From Table 1, it can see that the needs of recursion are in different areas. The method given by (Le Mortellec et al., 2013) and (Esmaeili et al., 2017) are adapted to fixed systems, however, it is hard to apply for dynamic application like CM. Although the aggregation notions (encapsulation and composition) and the interactions between agents were given by (Sallez et al., 2015), the authors did not detail the aggregation mechanism. (Vo et al., 2012) provide a recursive agent metal-model, in which the aggregation mechanisms (capture and release) are discussed. However, the interactions between agents located at the same level or different levels are not detailed. MAS-R (Occello et al., 2019) presents a recursive agent model, details interactions and aggregation mechanisms for a dynamic architecture. This method seems to be suitable for CM applications. In following sections, we try to apply this approach to CM context. The adapted agent model, recursive architecture and aggregation mechanisms are detailed for the CM application.

3. CM RECURSIVE AGENT MODEL

In this section, we adapt the MAS-R agent model for CM application where we add energy and some other information to the model. The CM agent model consists in five parts:

- The *Knowledge* part of agent (K): contains the agent's ID, energy level, recursion state, abstraction level, dynamic functions (it is used to calculate recursion state), energy model (it records energy consumption of its components). Cooperative Agent List (CAL) and other basic information;
- The *Composition* part (C): contains the composition function to construct a higher level;
- The *Decomposition* part (D): contains the decomposition function which asks parent agent to destruct the abstraction level;
- The *Observer* (O): observes the recursion state in D to call the recursive (composition or decomposition) functions and the communication functions to update neighbor information if necessary;
- The *Recursive interaction* part (RI): contains the interaction functions with other agents in the same group or with agents on other levels (parents or children), data processing and state updating.

The *O* part of agent periodically applies dynamic functions on the *K* part to analyze the recursion state. Once it finds that aggregation or decomposition conditions are met, it runs the composition/decomposition functions to create a higher abstract level or remove the existed abstract level (composition and decomposition mechanisms are detailed in next section). Once the composition (or decomposition) process is done, the updating status process in the *RI* part will be performed to update the information stored in *K*.

Agents can be regrouped as a composed CM (see Fig. 4). The real node coupled with a node agent is an elementary material holon. These node agents belong to the same composed agent as shown by link *1* in Fig. 4. They can discuss with agents located at the same level by classical message as the interaction link *2*. The abstraction link which present the link between an agent with its parent is illustrated as the link *3*. Node agents use the transformation functions in *RI* part to send information to the composed agent.

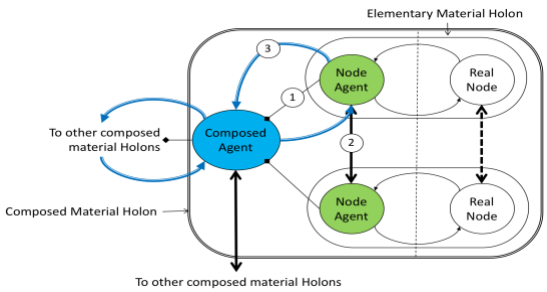

Fig. 4. Composed CM agent

4. CM RECURSIVE ARCHITECTURE

In different phases of the CM product lifecycle, one composed agents may cooperate with others. The recursive architecture is presented in this section to improve energy efficiency and extend service time.

The CM recursive architecture is shown in Fig. 5. In this example, there are six CNs in the physical part. Each node is represented by an elementary agent (Level 0) in the digital part. For an agent CM_i^j, *i* and *j* represent the abstraction level and ID respectively. Elementary agents (from CM_0^1 to CM_0^6) are aggregated into two CM agents (CM_1^1 and CM_1^2) on level *1* and CM_2^1 on level *2*. The agent located at the highest level has the global view of all CM products.

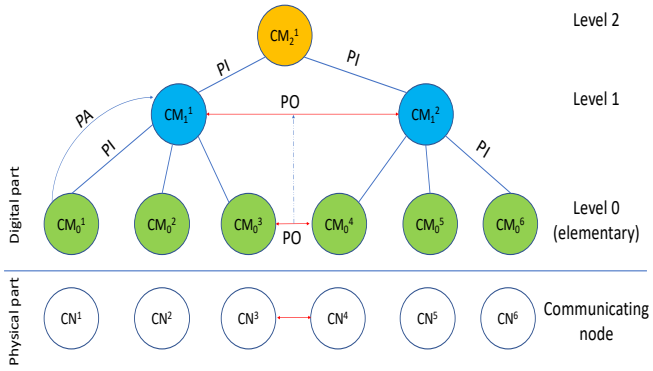

Fig. 5. The CM recursive architecture

There are three operators for aggregation, information exchange between agents in this recursive architecture:

- the *PA operator (Process Agent):* is responsible for the composition and decomposition processes.
- the *PI operator (Process Interaction)*: is defined as the transformation of an interaction between two abstract levels, for example, the level *n - 1* and *n*.
- the *PO operator (Process Organization)*: is defined as the transposition of relationship between levels. It can transfer the relationship from lower levels to higher levels.

When the observer part of an agent (for example CM_0^1 in Fig. 5) finds that composition conditions are met, it applies the *PA* operator to begin the composition process. The sequence diagram of composition process for generating the parent agent CM_1^1 is illustrated in Fig. 6. There are three elementary agents on level *0*. Composition process exchanges can be explained in five steps:

1. The CM_0^1 creates a parent agent CM_1^1 and adds itself to the CAL of CM_1^1 (message *1* in Fig. 6).
2. The *request_compose* messages are then sent to all cooperative agents in its CAL (message *2* and *3*).
3. The invited agents process the composition request and reply an *accept_compose* message to accept invitation (or a *refuse_compose* message to refuse). CAL is also included in the *accept_compose* message (message *4* and *5*).
4. The agent CM_0^1 adds accepted agents to CAL and sends the energy information to its parent node (message *6*). It then broadcasts *update_compose* messages to its child agents (message *7* and *8*).
5. Once received the *update_compose* message, the accepted agents update their own CAL to ensure that they have the same parent.

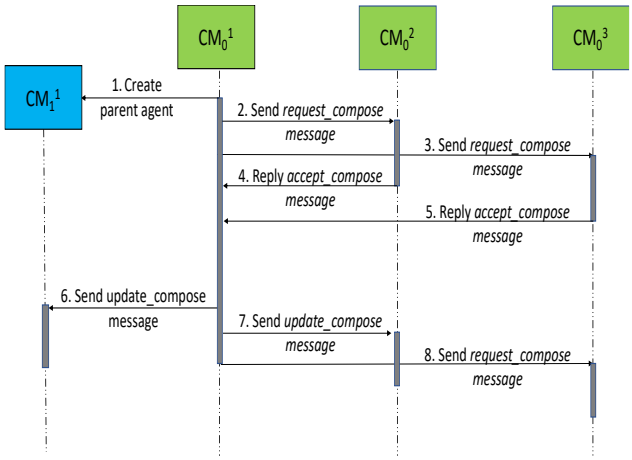

Fig. 6. The sequence diagram of composition

Once CM_1^1 is created and the structure is stable, it begins the process of energy model construction. This composed agent CM_1^1 sends *request_energy* messages to all children to get energy model information.

During the CM_1^1 lifecycle, its child agents check periodically whether they have relationship with another group (such as CM_0^3 can cooperate with CM_0^4 belonging to another CM). In that case, the agent CM_0^3 requests the parent information of agent CM_0^4. This information is then transferred to its parent agent CM_1^1 to update its CAL. By this way, the lower level is transferred to a higher level (the PO operator). The composition process will be repeated at CM_1^1 or CM_1^2 when composition conditions are met.

Once all composition process and energy model construction are completed as in Fig. 5. The agent at the highest level (the agent CM_2^1) can ask current energy level of child agents to estimate CM lifetime. The estimation process example is shown in Fig. 7.

The agent CM_2^1 first send *request_energy* messages to its child agents (CM_1^1 and CM_1^2). These agents then check whether they have current energy level information. They can reply it directly if they have (such as the aggregation agent CM_1^2 in this example). Otherwise, the agent (CM_1^1) asks all child agents for their energy information thanks to the PI operator. Once received all replies, the composed agent (CM_1^1) can aggregate information and reply an aggregated *ack_energy* message (minimal or average energy level) to parent agent CM_2^1. At the end, the agent CM_2^1 can then estimate CM lifetime based on the recent energy levels and energy consumption model.

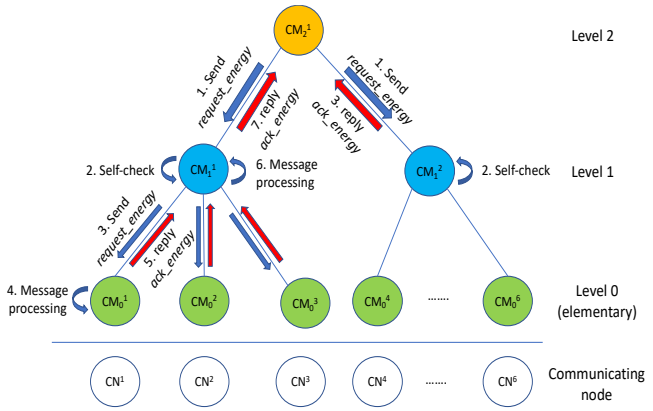

Fig. 7. CM lifetime estimation process

If the agent CM_0^3 has run out its energy. Once the observer of a cooperative agent (for example CM_0^2) notices the disappearance of this agent, the decomposition process begins as following:

1. The agent CM_0^2 sends a *request_reduction* to its parent agent CM_1^1.
2. The agent CM_1^1 processes the request and updates CAL, *update_reduction* messages are then sent to the agents in CAL.
3. The child agents update their own CAL based on the received *update_reduction* message.

To resume this section, the MAS-R adaptation for CM application is presented. The composition and decomposition mechanisms for constructing or destroying abstraction levels are detailed. In addition, the residual energy estimation process is also given for recursive architecture. In next section, the usefulness of this architecture is demonstrated on an applicative example.

5 APPLICATIVE SCENARIO

In some phases such as the construction for concrete element, two aggregated CM products may cooperate to improve energy performance as shown in Fig. 8. Each aggregated element (green and red agents) is composed by three elementary agents. The cooperation between the two independent DTs is discussed hereafter.

Once the observer part of an agent (CM_0^3 in the green part) detected the red part, it sends *request_coop* to warm its parent. The parent agent (CM_1^1) decides to cooperate or not and sends the *ack_coop* to all children. There are three cooperation possibilities (refuse, association and aggregation). The first case is no relation. Once the elementary agents receive a refuse message, they will maintain the original working state.

If the parent agent finds that the red part is related, it then replies an association message to all children. Only the connected agents (CM_0^3 in green and CM_0^4 in red) can exchange messages with each other. The PO operator can transfer this relation between the connect agents to its parent agents (CM_1^1 in green and CM_1^2). Therefore, these two parts can always know each other's states and cooperate to accomplish specific tasks if necessary.

Once the recursive states are met, all elementary agents will stop activity and wait orders from their parent node. The composed agent (CM_1^1) then begins the composition process thanks to PA operator as in Fig. 6. Once the composition process and energy model construction are completed, the agent CM_2^1 can then estimate CM lifetime as that in Fig. 7.

6. CONCLUSION AND PERSPECTIVES

This work presents the communicating material concept which is a special intelligent product with composition and decomposition abilities. Recent works applying MAS recursion are reviewed and compared with four characteristics

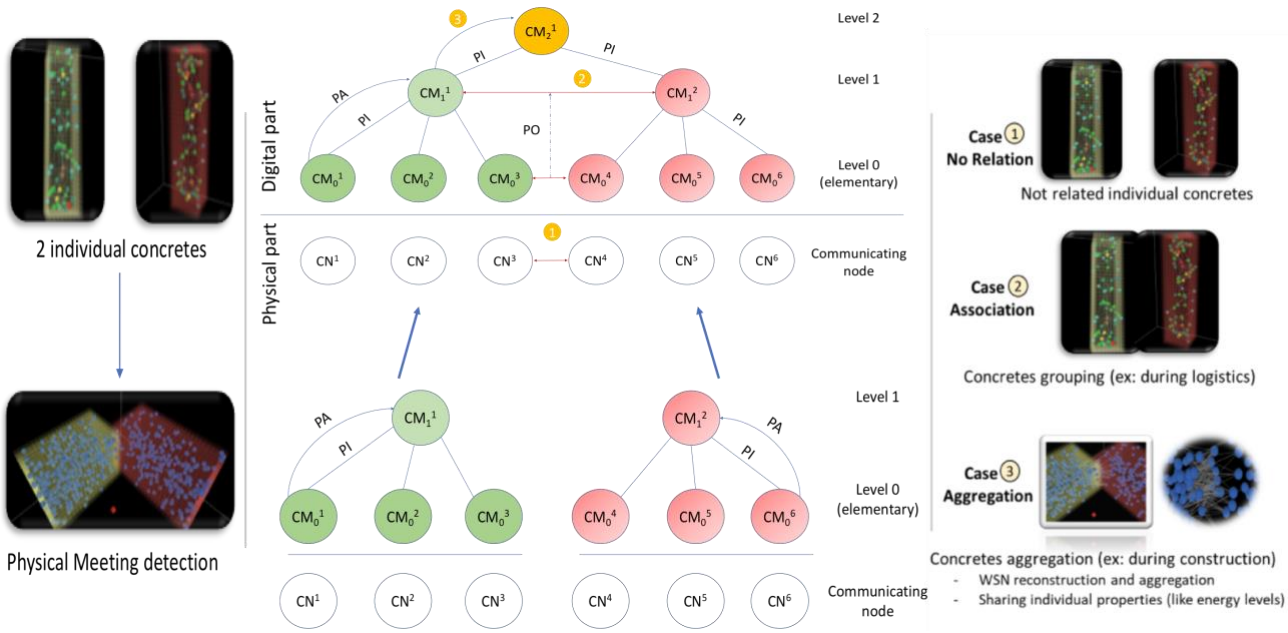

Fig. 8. Cooperation of two aggregated agents

to find an appropriate approach for CM application. The recursive architecture MAS-R proposed by (Occello et al., 2019) is quite proper for the communicating material. The recursive architecture can meet the dynamic requirement of CM application. However, the intelligence distribution of agents on different levels are not discussed in MAS-R. We apply MAS-R approach and develop the missing parts needed for CM energy management. CM agent model and the recursive MAS architecture with composition/decomposition mechanisms are presented. For CM application, the agent located at highest level may estimate the residual lifetime and evaluate different strategies to provide an optimal solution to extend network lifetime. Applying the MAS-R method to CM energy management seems to be promising as illustrated on the CMs interaction scenario.

In the following works, the composition/decomposition conditions as well as the dynamic functions will be specified. The interactions between agents on different levels will be developed. Besides, the intelligence at agents like data aggregation, residual energy estimation, selection of different data collection strategies will be specified. We will then apply the design method to the McBIM project, and develop a recursive digital twin to manage the energy and monitoring data of sensor nodes in real concretes.

ACKNOWLEDGMENTS

Authors thank financial support from the French National Research Agency (ANR) under the McBIM project, grant number ANR-17-CE10-0014.

REFERENCES

Derigent, W., David, M., WAN, H., Dragomirescu, D., Takscs, T., Gael, L., Roxin, A., Melet, R., and Montegut, L., (2019) 'Materials communicating with the BIM: Aims and first results of the MCBIM project', in *Proceedings of the 12th International Workshop on Structural Health Monitoring*, Standford, USA, 2019.

Esmaeili, A., Mozayani, N., Motlagh, M., and Matson, E., (2017) 'A socially-based distributed self-organizing algorithm for holonic multi-agent systems: Case study in a task environment', *Cognitive Systems Research*. Elsevier B.V., 43, pp. 21–44.

Gabor, T., Belzner, L., Kiermeier, M., Beck, M. and Neitz, A.. (2016) 'A Simulation-Based Architecture for Smart Cyber-Physical Systems', in *IEEE International Conference on Autonomic Computing (ICAC)*. Wurzburg, Germany.

Grieves, M. and Vickers, J. (2016) 'Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems', *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, (March), pp. 1–327.

Koestler, A. (1968) 'The ghost in the machine'. Hutchinson.

Kubler, S., Derigent, W., Thomas, W. and Rondeau, E. (2010) 'Problem definition methodology for the "communicating material" paradigm', *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 10(PART 1), pp. 198–203.

Meyer, G. G., Framling, K. and Holmstrom, J. (2009) 'Intelligent Products: A survey', *Computers in Industry*, 60(3), pp. 137–148.

Le Mortellec, A., Clarhaut, J., Sallze, Y., Berger, T. and Trentesaux, D. (2013) 'Embedded holonic fault diagnosis of complex transportation systems', *Engineering Applications of Artificial Intelligence*. Elsevier, 26(1), pp. 227–240.

Negri, E., Fumagalli, L. and Macchi, M. (2017) 'A Review of the Roles of Digital Twin in CPS-based Production Systems', *Procedia Manufacturing*. The Author(s), 11(June), pp. 939–948.

Occello, M., Jamont, JP., Ben-Yelles, C.B. (2019) 'A multi-level generic multi-agent architecture for supervision of collective cyber-physical systems', *International Journal of Autonomous and Adaptive Communications Systems*, 12(2), pp. 109–128. doi: 10.1504/IJAACS.2019.098590.

Rosen, R. Wichert, G.V., Lo, G. and D.Bettenhausen. K (2015) 'About the importance of autonomy and digital twins for the future of manufacturing', *IFAC-PapersOnLine*. Elsevier Ltd., 28(3), pp. 567–572. doi: 10.1016/j.ifacol.2015.06.141.

Sallez, Y., Montreuil, B. and Ballot, E. (2015) 'On the activeness of physical internet containers', *Studies in Computational Intelligence*. Elsevier B.V., 594, pp. 259–269. doi: 10.1007/978-3-319-15159-5_24.

Vo, D. A., Drogoul, A. and Zucker, J. D. (2012) 'An operational meta-model for handling multiple scales in agent-based simulations', *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future,*. doi: 10.1109/rivf.2012.6169849.

Wan, H., David, M. and Derigent, W. (2020) 'Energy-efficient chain-based data gathering applied to communicating concrete', *International Journal of Distributed Sensor Networks*, 16(8). doi: 10.1177/1550147720939028.

Wong, C. Y., McFarlane, D., Ahmad Zaharudin, A. and Agarwal, V. (2002) 'The intelligent product driven supply chain', *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 4, pp. 393–398. doi: 10.1109/icsmc.2002.1173319.