International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 6, December 2023, pp. 6228~6239
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i6.pp6228-6239

Homogenous and interoperable signaling computer interlocking through IEC 61499 standard

Ikram Abourahim, Mohsine Eleuldj, Mustapha Amghar
SIP Laboratory, Mohammadia School of Engineers, Mohammed V University in Rabat, Rabat, Morocco

Article Info

Article history:

Received Mar 27, 2023
Revised Jun 6, 2023
Accepted Jun 22, 2023

Keywords:

Computer interlocking
Distributed system
IEC 61499 standard
Interoperability
Railway signaling
Service oriented architecture

ABSTRACT

The technological evolution of signaling systems has created a dependency from infrastructure managers to suppliers and industrials dominating the market. Indeed, for each deployed computer interlocking, the modification of field equipment is required to allow an adaptation with the new interlocking in terms of communication protocols and logical interface. In addition, to ensure safe traffic of trains, the communication of railway signaling data is necessary between interlockings. However, delayed deployments from one station to another make the establishment of communication channels costly and difficult, or even impossible, since each supplier keeps confidential its communication protocols and usually opts for interfacing based on wired logic. This paper presents our approach to a homogeneous architecture of interlocking meeting modularity requirements, interoperability, and logical interfacing between interlockings. This approach relies on a classification of internal functions of the computer interlocking, a distribution of the execution of those functions and making useful information available for interfaces between adjacent interlockings through the IEC 61499 standard coupled with service-oriented architecture (SOA).

Corresponding Author:

Ikram Abourahim
SIP Laboratory, Mohammadia School of Engineers, Mohammed V University in Rabat
Rabat, Morocco
Email: ikramabourahim@research.emi.ac.ma

1. INTRODUCTION

The digitalization of the industry has created new challenges for managers and decision-makers. Industry is naturally composed of a set of heterogeneous technologies whose interconnection makes it possible to constitute production lines or automated control and supervision systems. Most often, the data exchanged between the different devices is neither standardized nor controlled.

The race for new manufacturing processes and technologies continues to accelerate [1]. To bring ever more value to industrial customers and end users, the manufacturer must constantly review it is copy and enrich its technical assets [2]. Interoperability is one of the development challenges of tomorrow's industries as introduced in [3] and [4] whether on the Software [5] or Hardware side [6].

In the context of Industry 4.0, the challenge for the manufacturer is to create a dialogue between the elements of the production tool from various horizons. Connected objects and the internet of things (IoT) [7] are invading our daily lives and those of companies. The market offers a plethora of technical solutions, more or less proprietary.

The collaboration between equipment allows the manufacturer to gain agility, when changing products or modifying manufacturing processes. The time to market a new product depends on the execution time of the production line and the machine downtime, which in turn depend on the coupling between the components and the agility of the maintenance process. As a result, thanks to digitalization, both times can be significantly reduced [1], [2].

In addition, productivity can be improved by accurately measuring data from tools, tuning production parameters in real time and alerting to the risk of failure. The data retrieved makes it possible to optimize and make production more flexible. Quality, traceability and therefore customer satisfaction are also the big winners of interoperability by offering a whole range of information on the manufacturing process.

Like any connected system, interoperability tools entail a major risk of loss of control over the data and information that passes between devices [8]. Similarly, the centralization of digital data and their increasingly complex reprocessing require expertise that generally goes beyond the company's field of expertise. As with the entire information system, the implementation of a move towards a more cooperative and interoperable system requires a great deal of vigilance and prior validation.

Several standards [9] and studies [10], [11] have been interested in responding to the need for interoperability, especially with the inclusion of IoT in the industrial field [7] and its opening to the cloud [12]. Among the standards that have proven their effectiveness for interoperability in the industrial field are the standard IEEE 1451 for smart sensors [13], [14], the open platform communications unified architecture (OPC UA) protocol imposing security criteria in the exchange, storage or dissemination of data [15], [16], and the standard IEC 61499 [17]–[19] for distributed command and control systems.

In signaling railway system, the interoperability is also an issue [20] due to the non-homogeneity of information technology (IT) solutions and the different interfaces between systems of different suppliers deployed. Indeed, technological solutions insure a multitude of facilities and services for managing train traffic. As solutions, we consider remote control and monitoring of installations, computer interlocking, and automatic driving.

Thus, to deal with the problem of interoperability between signaling computer interlocking, we propose a standardization of the exchange between computer interlockings through the IEC 61499 standard. In this paper, we present firstly the IEC 61499 standard and their benefits. After that, we carried out our proposition of a functional model for railway signaling interlocking whose execution is distributed in an architecture of calculators. This third section presents, as well, the services approach chosen for better flexibility and interoperability between interlockings. Finally, the discussion and analysis of results lead to the conclusion and perspectives of our research.

2. IEC 61499 STANDARD: DISTRIBUTED INDUSTRIAL MEASUREMENT AND CONTROL SYSTEMS

IEC 61499 is an international standard for function blocks used in distributed industrial process measurement and control systems. It was first published in 2005 and revised in 2012. It is specification [21] defines a general model of a control system, divided into four parts as seen in Table 1. Therefore, IEC 61499-1 defines an open architecture of function blocks for distributed integrated control and automation systems. This open architecture is combined with appropriate IEC 61499-4 compliance profiles and software tools to meet IEC 61499-2 requirements.

Table 1. IEC 61499 standard

Parts	Application domain
IEC 61499-1	Architecture [17]
IEC 61499-2	Requirements of the software tool [18]
IEC 61499-3	Application guide (withdrawal in 2008)
IEC 61499-4	Rules for compliance profiles [19]

As a result, reusable software modules (functional blocks) can be developed and deployed in distributed systems that will meet the requirements of portability, interoperability and configurability as seen in Figure 1. The first requirement ensures that software tools correctly accept and interpret software components and system configurations produced by other software tools. The second one allows embedded devices work together to perform the functions needed by distributed applications. Finally, the third requirement facilitates the configurability of any device and its software components by software tools from different vendors and suppliers.

In fact, IEC 61499 has proven its ability to manage effectively the distribution of system functions as well as its agility during deployment, maintenance, or upgrade systems as mentioned in [14], [22]. This ability is due to the requirements' standard and also their main advantages. The advantages of this standard are encapsulation, reuse, distribution, and integration.

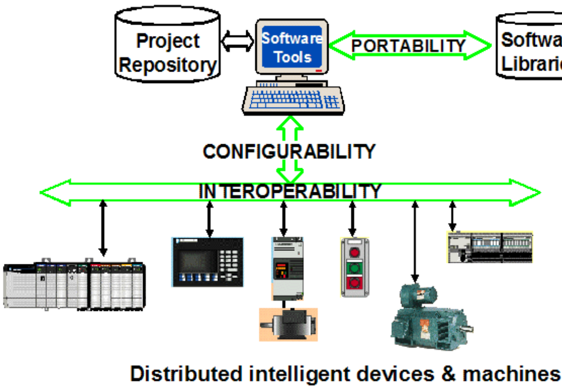

Figure 1. IEC 61499 requirements

2.1. Encapsulation

In IEC 61499, the basic unit for encapsulating and reusing basic functions is a function block. A class defines the behavior of multiple instances in an object-oriented manner. It includes event input and output and data input and output to achieve synchronization between data transfer and program execution in distributed systems.

2.1.1. Basic functional block

The basic functional block types as shown in Figure 2 are the atoms that build higher-level molecules. Software developers can encapsulate intelligent properties, by using IEC 61499-2 compliant software tools, as algorithms written in Java or IEC 61131-3 programming languages. The execution of these algorithms is done by execution control cards (ECCs) as shown in Figure 3, which are event-based state machines similar to Harel's state diagram [23].

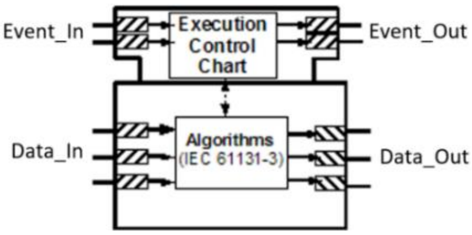

Figure 2. Basic functional block [17]

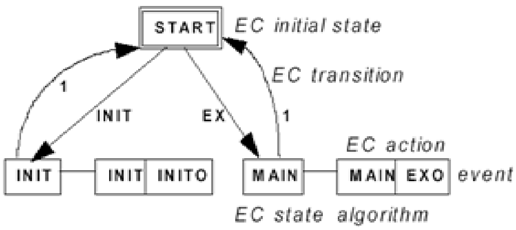

Figure 3. Execution control card [17]

2.1.2. Service interface functional block

Another type of atomic function block is the service interface function block (SIFB) type. This represents an interface to low-level services provided by an embedded device's operating system or hardware. These services can be graphical user interface (GUI) elements (cursors, buttons, or lights) or services communicating with external systems or interfaces to hardware (temperature sensors, engine speed controllers, control valves, or ambient light intensity controllers).

The IEC 61499-compliant software tools and associated runtime packages provide an extensive GUI and communicate with the SIFB as shown in Figure 4. A hardware SIFB provider (usually an embedded device manufacturer) can use an IEC 61499-compliant software tool to document its operations in the form of a service sequence diagram, as shown in Figure 4 on the right. The graph then resumes the expected exchange between client and server.

2.2. Reuse

Software developers can use IEC 61499-2 compliant software tools to create higher-level function block molecules (called composite function block types) from lower-level function block atoms that can, in turn, be simple or compound. To do this, the event and data interfaces of the composite type need to be specified and then populated with a diagram showing how the functional blocks of its internal components are connected. In this type of function block, the execution of the algorithm in the component function block is controlled by the flow of events from one component to another, as shown by the links in Figure 5. This combination of function blocks enables defined function blocks (basic or function blocks) to be reused. Composite if required by other functions.

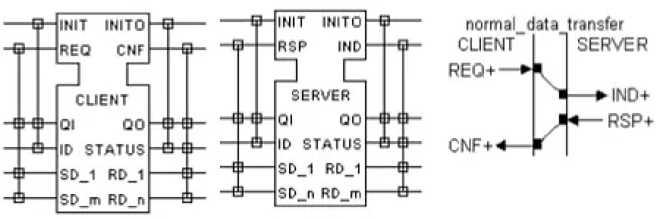

Figure 4. Service interface functional block [17]

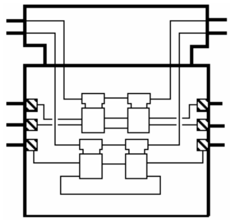

Figure 5. Composite functional block [17]

2.3. Distribution

In the IEC 61499-1 architectural model, distributable operations are erected by hitching cases of applicable functional block types (basics or mixes) with applicable event and data connections. Using IEC 61499-2 biddable software tools, these structure blocks can be distributed to physical bias over a network, as long as these biases misbehave with the applicable compliance profile (19). Figure 6 shows the possibility of distributing operations (operation A, App.B, App.C) to multiple bias where the functional blocks are executed as shown at the top of Figure 6. It is also possible to distribute an operation across multiple coffers within a device. coffers can be multiple processors connected to a backplane or multiple tasks within a single multi-core processor.

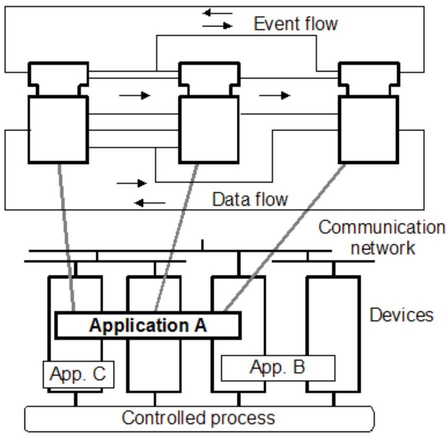

Figure 6. Distribution-IEC 61499 [17]

2.4. Integration

In the IEC 61499 armature, coffers give the services demanded to integrate all operations into a functional distributed system. Indeed, software tools biddable with IEC 61499 grease integration through numerous installations. Firstly, mapping dispatches that are transmitted between bias in input and affair events and data in SIFB. also, using event and data inputs and labors to spark the performance of introductory and compound functional block algorithms, and attend their operation with other functional blocks. Finally mapping the inputs and labors of data and events from I/O SIFBs to the inputs and labors of the system, where it can describe what is passing in the physical world and take applicable physical action in response.

3. DESIGN OF HOMOGENEOUS AND INTEROPERABLE COMPUTER INTERLOCKING

The continuous evolution of the technology inspires different industrial in the field of railway signaling but lead as well to the lack of homogeneity between interlockings proposed and then to heavy investments in each modification or upgrade of the railway installation. Moreover, the interface between interlockings of adjacent stations remains a handicap and is managed most of the time by electromechanical interfaces even if computer interlockings are deployed. This situation is due to the non-standardization of the communication protocol and to the difference in the processing of the information received. Therefore, we design a homogeneous architecture and a logic interface between computers interlocking on borders.

3.1. Functional model

In our project, we are primarily interested in the harmonization of computer signaling interlockings. Thus, we propose an architecture based on functional blocks according to the IEC 61499 standard. The modeling of the overall system using the SysML block diagram is shown in Figure 7.

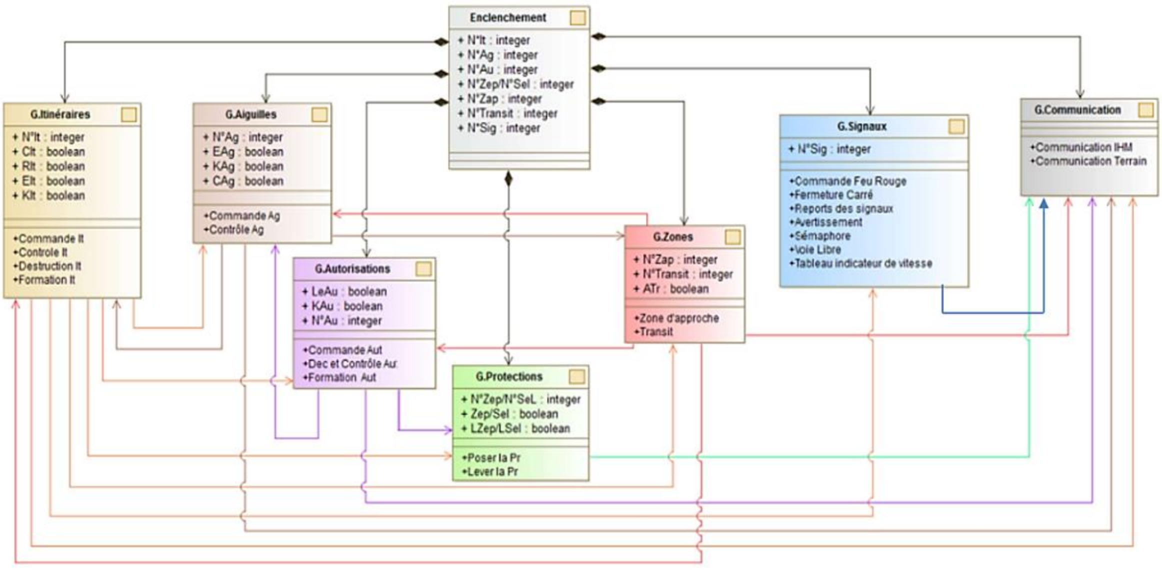

Figure 7. Functional model-SysML [23]

This model is based on Moroccan signaling principles but as a global model, it is applicable for different signaling principles of other railway infrastructure managers with adaptability in internal functional blocks. Each block is linked to a family of functional blocks that run in interaction and have attributes in common [24]. To execute this functional model, we propose a distributed architecture of calculators.

3.2. Architecture of distributed calculators

To realize the interoperability solution through functional blocks, we are considering a new distributed architecture proposal for the signaling interlocking. Indeed, in the centralized architecture, the central computer (level 1 in Figure 8) is linked to field equipment through the object controllers (OCs) that are represented at level 2 in Figure 8. Each type of field equipment has an appropriate OC. However, to realize the interoperability proposal via functional blocks, we are considering a new architecture for signaling computer interlockings. In summary, the interlocking, composed of a single computer according to the centralized architecture, will be broken down into a central calculator and auxiliary calculators according to the distributed architecture as summarized in Figure 9.

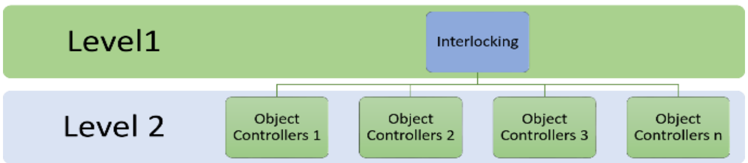

Figure 8. Centralized interlocking architecture

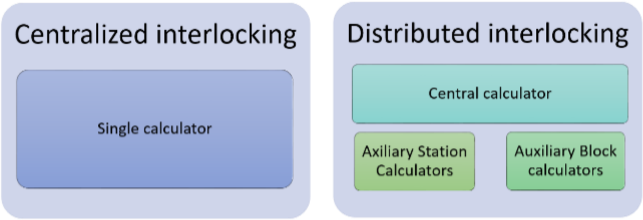

Figure 9. Decomposition of interlocking

IEC 61499 functional blocks allow a system to be decomposed into elementary functions that can be executable in a distributed environment with synchronous logic as shown in Figure 6. Therefore, in our proposed configuration, the interlocking of a station is composed by a network of sub-calculators. The main functions are performed on the central calculator, the functions related to the station field equipment are distributed on auxiliary station computers and only the necessary information is sent to the central computer. Finally, the functions related to the field equipment located on the open line (between the stations) are executed on auxiliary block computers and only the useful information, for the supervision or the command of tracks, is sent to the central calculator as shown in Figure 10.

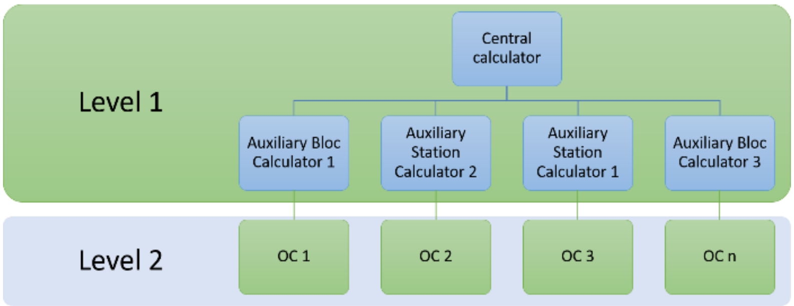

Figure 10. Distributed interlocking architecture

In addition, the calculation of the functions initially performed on the computer interlocking will be distributed between the central calculator and the auxiliary calculators as shown in Figure 10. In fact, the main functions are performed on the central computer. Then, the auxiliary station calculators execute the functions related to the field equipment at the auxiliary station calculator and only the necessary information is sent to the central computer. For example, if we consider the station in Figure 11, the station will be divided into 2 parts on the left and right sides, and then the equipment of each part is linked to the auxiliary station computer on the right or left. Finally, the auxiliary block calculators execute the functions related to the open line equipment and only the necessary information is transmitted to the central computer.

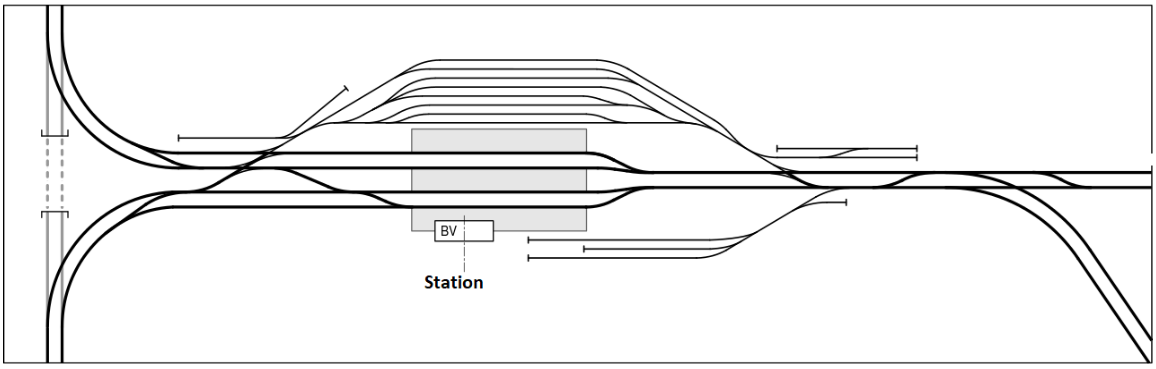

Figure 11. Example station map

3.2.1. Time analysis

Indeed, the Interlocking calculator operates on a cyclical basis. Cycle time (t_{cycle}) is composed of function execution period (t_{exe}) and the output data sending period (t_{com}) as modelled in (2). At the beginning of the cycle, all available input data is read (i.e., the status of field equipment and controls from the supervisory station). Then, all the interlocking functions are executed successively. At the end of t_{exe}, the outputs (commands for object controllers and indications for the supervisory station) are sent from the processing unit to the communication unit for compression and transmission. Figure 12 illustrates the concept. Thus, the analysis in next sections focuses on the influence of architecture choices on t_{exe} and t_{com}.

$$t_{cycle} = t_{exe} + t_{com} \quad (1)$$

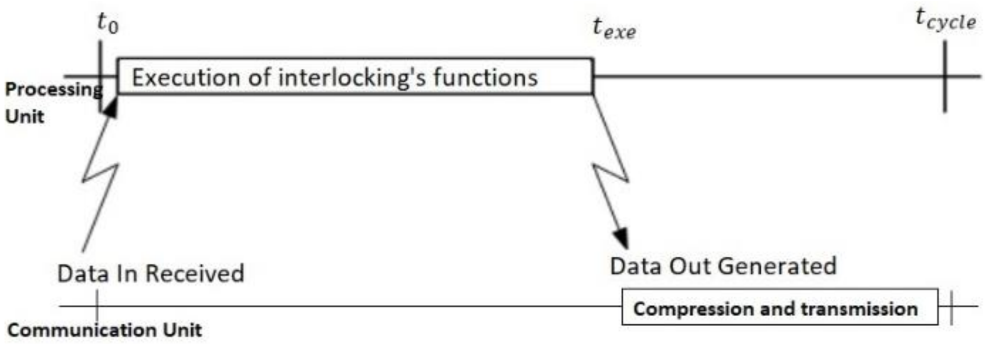

Figure 12. Illustration of the interlocking cycle time

3.2.2. Execution time

Based on Moroccan railway signaling principles, we have established a new distribution of interlocking's functions. Instead of calculating all the functions at the level of a single interlocking computer, we chose to distribute them between the central calculator on one hand and the auxiliary stations and blocks calculators on the other hand as shown in Figure 10. Each function is related to an execution time that affects the overall execution time of the interlocking modelled in (2). Thus, reducing the number of functions performed by a calculator reduces the execution time in a cycle for each calculator and then ensures overall optimization at the level of the interlocking cycle period [25].

$$t_{exe} = \sum_{i=1}^{N} time(f_i) \quad (2)$$

where, t_{exe} is cycle execution time, $time(f_i)$ is execution time of function f_i, and N is the total number of functions.

3.2.3. Communication time

For the execution of the interlocking functions, exchanging variables' states is essential to enable overall system execution with consistency and logical synchronization. Each family of functions has variables to exchange (data in/data out) as shown in Figure 13. When using centralized architecture, all the information collected by the object controllers is sent to the interlocking (single calculator Figure 8). However, executing the same functions in distributed architecture allows sending only results of functions performed in the auxiliary calculators to the central computer as shown in Figure 10.

To quantify the relevant of our proposal, we choose the category of signal to compare the data exchange flow between centralized and distributed architectures because the functions in this category are fully computed on an auxiliary block calculator. For internal variables of signal functions, we can significantly reduce the data exchange flow from the central calculator to the object controllers when the calculation is performed on the auxiliary block calculator, so that the exchange of 56 variables in the centralized architecture as shown in Figure 14 is reduced to 0 variables in the distributed architecture as shown in Figure 15. In addition, it reduces the flow of data to the central computer from 45 variables sent by the object controllers as shown in Figure 14 to 23 variables sent from the auxiliary block computer as shown in Figure 15. Thus, if we consider a linear model (3) for communication time, reducing the number of variables exchanged reduces communication time.

$$t_{com} = a + \frac{K_{CA}}{b} \quad (3)$$

where, t_{com} is communication time, a is latency, b is debit, and K_{CA} is number of variables (data out of functions (f_i) transmitted by the auxiliary calculator).

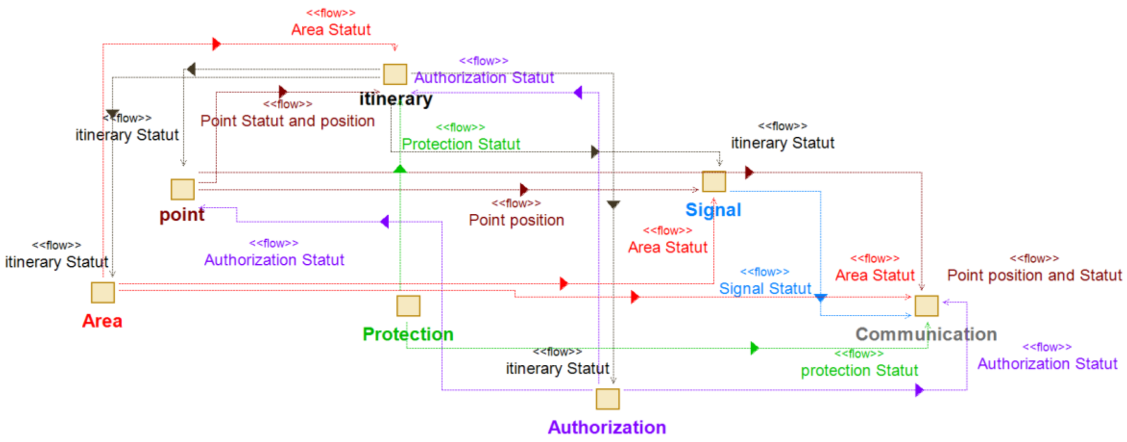

Figure 13. Diagram of interchange between interlocking's functional blocks [25]

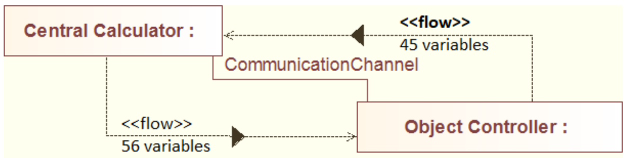

Figure 14. Signal data exchange-centralized architecture

Figure 15. Signal data exchange-distributed architecture

3.3. Services approach for interlocking interfaces

In the continuity of the functional improvement of the interlocking module of the signaling system and responding to the need for interoperability and exchange between adjacent interlockings, we proposed an interface between interlockings with a service-oriented approach. On one hand, IEC 61499 standard defines a form of service interface functional block (SIFB). The SIFBs are the elements made available for interfacing between calculators. In the other hand, the proposal of a service-oriented architecture (SOA) to frame the exchange of SIFB ensures the provision of essential services for the exchange in a perspective of independence between interlockings.

By joining the sense of the SIFB according to IEC 61499, the information transmitted is automatically fed by/into the SIFB concerned by the communication between adjacent interlockings. Moreover, when this communication is related only to field equipment's status, the concerned variables are sent directly from the auxiliary calculator (station or block) to the relevant SIFB and does not impact the execution cycle of the central calculator. As well, the coupling of SIFB with SOA architecture services facilitates interfacing adjacent interlockings when deploying new interlocking or during delayed deployments (years of difference between station commissioning).

3.3.1. Interlocking interface functions

To ensure the communication of adjacent interlockings, we rely on feedback from the deployment of the European rail traffic management system (ERTMS) system to define the SIFB. Indeed, ERTMS messages give a movement authority (MA) which depends on the appearance of one or more successive signals. The content of the MA is different depending on the type of signal; block or station.

Also, the perimeter of the interlocking is geographically limited by interface signals that are closely related to the boundary areas as shown in Figure 16. We therefore chose to link the signal and area functional blocks to the SIFB. For this, there are two types of functional blocks for the signal interface; block signal functional block that ensures the safety of successive train movements on a full line and Signal Station functional block that ensures the safe movement of trains in stations. For the area limit, The SIFB in connection with the functional block linked to the boundary zone as shown in Figure 16 provides information on the occupation or not of the area located within the geographical boundaries of the interlocking control.

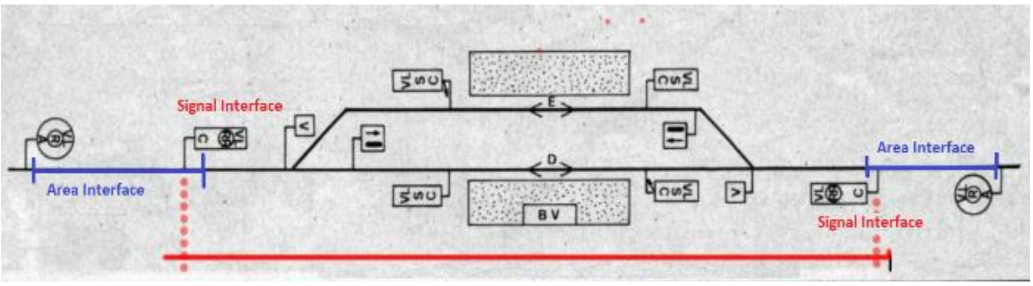

Figure 16. Elements of interfaces in stations-example

3.3.2. Interlocking interface services

We chose, in the first place, to carry out the modeling of interface services by SoaML. This modeling language allows us to dress the different levels of SOA architecture, composed of services, through their different diagrams (service interface diagram, participants diagram, service architecture diagram …). As service interface, we carried out two types of services that are needed to realize the interface between adjacent interlocking. Those services are *SignalStatut* and *AreaStatut* and they are related to the signal and the area within the borders of each interlocking. Participants diagram of those services is represented Figure 17 and explained in [26].

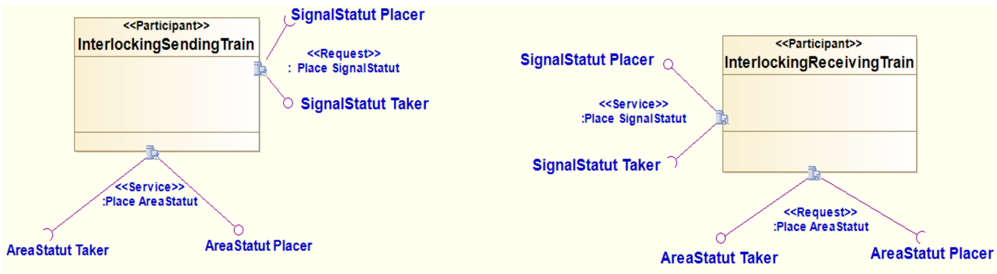

Figure 17. Participants' diagram

4. CASE STUDY

4.1. Simulation

To confirm the relevance of the functional model proposed, we carried out a simulation on ISaGRAF for a station example used for training at National Office of Railway in Morocco (ONCF). Interlocking functions of the simulated station are distributed in five simulated calculators as shown in Figure 18. We made, as well, a supervision interface using the ISaVIEW software. This interface allowed us to supervise and control the elements of the station studied from the signaling view as shown in Figure 19, thus giving the information concerning the status of each element represented (itinerary, point, area and signal). The interface also offers the possibility to simulate commands and simulate the condition of field equipment.

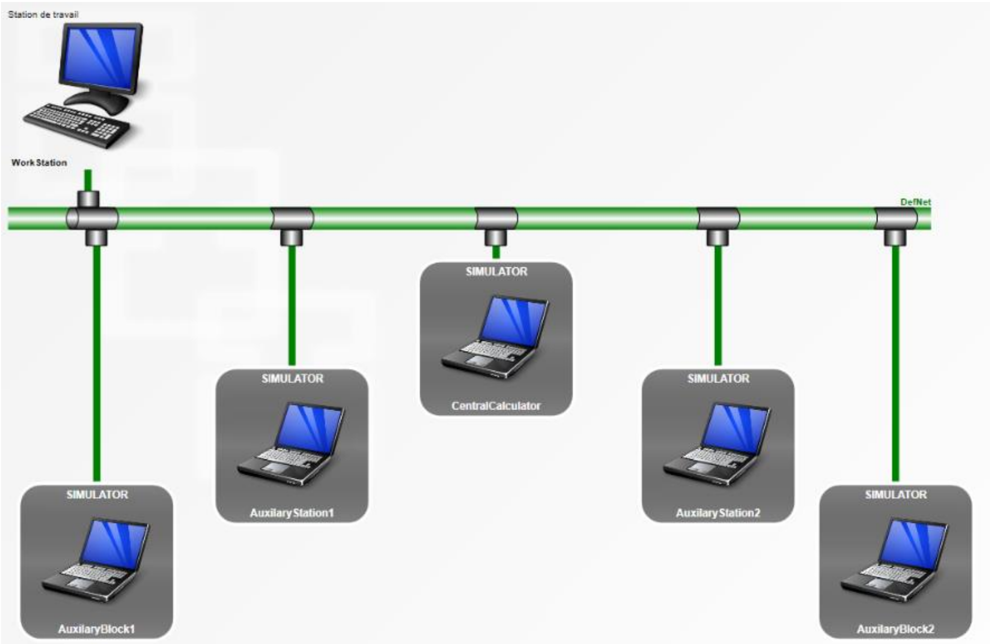

Figure 18. Distributed architecture-deployment network

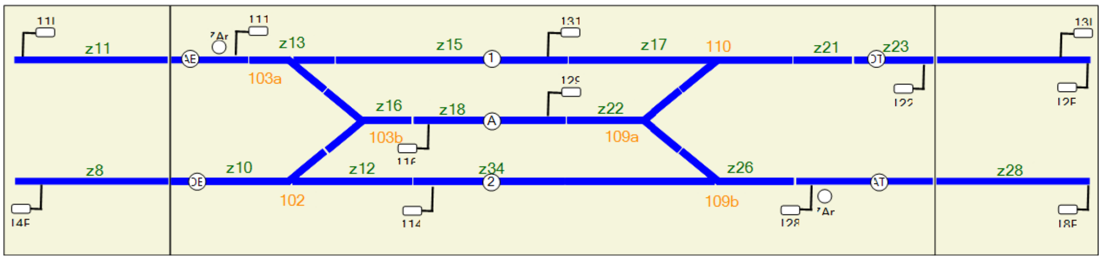

Figure 19. Simulation view

4.2. Results and discussion

When running the program, simulated supervision (workstation) does not make any distinction between data changes at the central calculator level and at the Auxiliary calculator level. The result is in accordance with the rules and principles of signage and the distribution remains transparent for the supervisory position. However, since we carried out the simulation with an educational version of ISaGRAF, a limitation of the number of links of a data (global variable) from the transmitter block (*Data_Out*) to consumer blocks (*Data_In*) required the implementation of a bypass.

Thus, we have chosen to set up communication blocks in the central computer between families of functions (itinerary, points, and signals). This solution can be retained for final deployment, as it facilitates the exploitation of the data exchanged and supports the modular feature desired when modifying interlockings. Indeed, when adding an itinerary, the related functional block can extract *Data_In* only from the internal communication block of the central computer without the need to establish a link with each Functional Block of other function families.

5. CONCLUSION AND PERSPECTIVES

Our proposal focuses on the architecture of computer interlockings railway signaling. On one hand, the homogenization of the functional model increases the performance of the internal functions of the interlocking. This homogeneity also brings modularity in the system and its preparation for interoperability and interfacing with adjacent interlockings. On the other hand, the distribution of the operation of the interlocking in a distributed architecture of programmable automatons, makes functions available and decouples interfacing.

As a result, the harmonization between interlockings will allow a mastery of the technology deployed by the infrastructure manager, flexibility during software changes following station facilities and a prearranged interface between interlockings not dependent on the provider. In addition, trials and tests are an essential phase in the signaling project management process and its control is regulated according to CENELEC standards. Thus, the development of a validation model to automatically test the final system is essential. The automatic test system will facilitate the management of test phases laboratories and sites of the signaling system to put it into operation safely.

ACKNOWLEDGEMENTS

The research work for this paper is the result of a collaboration between EMI and ONCF.

REFERENCES

[1] C. A. Garcia, G. Caiza, and M. V. Garcia, "Oil and gas upstream sector: the use of IEC-61499 and OPC," in *Handbook of Smart Materials, Technologies, and Devices*, Cham: Springer International Publishing, 2021, pp. 1–32.

[2] R. D. Bartusiak *et al.*, "Open process automation: a standards-based, open, secure, interoperable process control architecture," *Control Engineering Practice*, vol. 121, Apr. 2022, doi: 10.1016/j.conengprac.2021.105034.

[3] F. Ameri, D. Sormaz, F. Psarommatis, and D. Kiritsis, "Industrial ontologies for interoperability in agile and resilient manufacturing," *International Journal of Production Research*, vol. 60, no. 2, pp. 420–441, Jan. 2022, doi: 10.1080/00207543.2021.1987553.

[4] H. da Rocha, A. Espirito-Santo, and R. Abrishambaf, "Semantic interoperability in the industry 4.0 using the IEEE 1451 standard," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2020, pp. 5243–5248, doi: 10.1109/IECON43393.2020.9254274.

[5] S. C. Jepsen, T. Worm, T. I. Mork, and J. Hviid, "Industry 4.0 middleware software architecture interoperability analysis," in *2021 IEEE/ACM 3rd International Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT)*, Jun. 2021, pp. 32–35, doi: 10.1109/SERP4IoT52556.2021.00012.

[6] F. Lelli, "Interoperability of the time of industry 4.0 and the internet of things," *Future Internet*, vol. 11, no. 2, Feb. 2019, doi: 10.3390/fi11020036.

[7] A. Amjad, F. Azam, M. W. Anwar, and W. H. Butt, "A systematic review on the data interoperability of application layer protocols in industrial IoT," *IEEE Access*, vol. 9, pp. 96528–96545, 2021, doi: 10.1109/ACCESS.2021.3094763.

[8] A. P. Allian, F. Schnicke, P. O. Antonino, D. Rombach, and E. Y. Nakagawa, "Architecture drivers for trustworthy interoperability in industry 4.0," *IEEE Systems Journal*, vol. 15, no. 4, pp. 5454–5463, Dec. 2021, doi: 10.1109/JSYST.2020.3041259.

[9] T. Burns, J. Cosgrove, and F. Doyle, "A review of interoperability standards for industry 4.0.," *Procedia Manufacturing*, vol. 38, pp. 646–653, 2019, doi: 10.1016/j.promfg.2020.01.083.

[10] M. A. Inigo, A. Porto, B. Kremer, A. Perez, F. Larrinaga, and J. Cuenca, "Towards an asset administration shell scenario: a use case for interoperability and standardization in industry 4.0," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, Apr. 2020, pp. 1–6, doi: 10.1109/NOMS47738.2020.9110410.

[11] S. C. Jepsen, T. I. Mork, J. Hviid, and T. Worm, "A pilot study of industry 4.0 asset interoperability challenges in an industry 4.0 laboratory," in *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Dec. 2020, pp. 571–575, doi: 10.1109/IEEM45057.2020.9309952.

[12] G. Pedone and I. Mezgár, "Model similarity evidence and interoperability affinity in cloud-ready industry 4.0 technologies," *Computers in Industry*, vol. 100, pp. 278–286, Sep. 2018, doi: 10.1016/j.compind.2018.05.003.

[13] D. Wobschall, "IEEE 1451-a universal transducer protocol standard," in *AUTOTESTCON (Proceedings)*, Sep. 2007, pp. 359–363, doi: 10.1109/AUTEST.2007.4374241.

[14] R. Abrishambaf, H. Da Rocha, and A. Espirito-Santo, "IEC 61499 and IEEE 1451 for distributed control and measurement systems," in *IECON 2021-47th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2021, pp. 1–6, doi: 10.1109/IECON48115.2021.9589997.

[15] S. Cavalieri, "A proposal to improve interoperability in the industry 4.0 based on the open platform communications unified architecture standard," *Computers*, vol. 10, no. 6, May 2021, doi: 10.3390/computers10060070.

[16] T. Lyu, U. D. Atmojo, and V. Vyatkin, "On automatic generation of OPC UA connections in IEC 61499 automation systems," in *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*, May 2022, pp. 1–6, doi: 10.1109/ICPS51978.2022.9816861.

[17] AFNOR, "Building blocks-part 1: architecture," *National standards and national normative documents*, NF EN 61499-1, 2013, Accessed: Aug. 12, 2022. [Online]. Available: https://www.boutique.afnor.org/fr-fr/norme/nf-en-614991/blocs-fonctionnels-partie-1-architecture/fa174805/41301.

[18] AFNOR, "Building blocks-part 2: requirements for software tools," *National standards and national normative documents*, NF EN 61499-2, 2013, Accessed: Aug. 12, 2022. [Online]. Available: https://www.boutique.afnor.org/fr-fr/norme/nf-en-614992/blocs-fonctionnels-partie-2-exigences-pour-les-outilslogiciels/fa175102/41320.

[19] AFNOR, "Building blocks-part 4: rules for conformance profiles," *National standards and national normative documents*. NF EN 61499-4, 2013, Accessed: Aug. 12, 2022. [Online]. Available: https://www.boutique.afnor.org/fr-fr/norme/nf-en-614994/blocs-fonctionnels-partie-4-regles-pour-les-profils-de-conformite/fa175509/42001.

[20] A. V Ozerov, "Towards safer rail control, command and signalling in the context of digitization," *Dependability*, vol. 20, no. 2, pp. 54–64, Jun. 2020, doi: 10.21683/1729-2646-2020-20-2-54-64.

[21] D. Drozdov, V. Dubinin, S. Patil, and V. Vyatkin, "A formal model of IEC 61499-based industrial automation architecture supporting time-aware computations," *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, pp. 169–183, 2021, doi: 10.1109/OJIES.2021.3056400.

[22] W. Dai, V. Vyatkin, J. H. Christensen, and V. N. Dubinin, "Bridging service-oriented architecture and iec 61499 for flexibility and interoperability," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 771–781, Jun. 2015, doi: 10.1109/TII.2015.2423495.

[23] D. Harel, “Statecharts: a visual formalism for complex systems,” *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, Jun. 1987, doi: 10.1016/0167-6423(87)90035-9.
[24] I. Abourahim, M. Amghar, and M. Eleuldj, “Interoperability of signaling interlocking and its cyber-security requirements,” in *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Apr. 2020, pp. 1–6, doi: 10.1109/IRASET48871.2020.9092173.
[25] I. Abourahim, M. Amghar, and M. Eleuldj, “Distributed architecture for interoperable signaling interlocking,” in *Smart Innovation, Systems and Technologies*, vol. 237, Springer Singapore, 2022, pp. 215–230.
[26] A. Ikram, M. Eleuldj, and M. Amghar, “Services interfaces for interoperability of signaling computer interlocking on borders,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 2640–2651, Jun. 2023, doi: 10.11591/ijece.v13i3.pp2640-2651.

BIOGRAPHIES OF AUTHORS

Ikram Abourahim received an engineering degree in telecommunication engineering from the National Institute of Posts and Telecommunication in Rabat (INPT), Morocco, in 2011. Currently, she is a Ph.D. Student at computer and production systems laboratory (SIP) from Mohammadia School of Engineers (EMI), Mohammed V University in Rabat, Morocco. Her research interests include the new generation of signaling systems, distributed automation systems, service-oriented architectures, communication protocols, and system interfaces. She can be contacted at email: ikramabourahim@research.emi.ac.ma.

Mohsine Eleuldj received his Ph.D. in 1989 at Universite de Montreal, Canada, and then joined, as a professor, the Computer Science Department at Ecole Mohammadia des Ingenieurs, Mohammed V University in Rabat, Morocco. His research interests include parallel processing, distributed computing, computer security, and hardware implementations. He can be contacted at email: eleuldj@emi.ac.ma.

Mustapha Amghar is a holder of a Ph.D. in industrial computing from Liege University in Belgium as well as a university habilitation in research from the Mohammadia School of Engineering in Morocco. He is currently a professor in the Electrical Department at Mohammadia School of Engineering. For over 30 years, he has been completing counseling missions for public and private institutions. His research interests include control systems engineering, electronic engineering, and aerospace engineering. He has also confirmed experiences in global navigation satellite systems, Galileo, global positioning systems, survey engineering, GPS navigation devices, EGNOS, localization, and tracking. He can be contacted at email: amghar@emi.ac.ma.