Available online at www.sciencedirect.com
ScienceDirect
IFAC-PapersOnLine 49-12 (2016) 249–254

A database-centric approach for the modeling, simulation and control of cyber-physical systems in the factory of the future.

Andrea Bonci*, Massimiliano Pirani*, Sauro Longhi*

**Dipartimento di Ingegneria dell'Informazione (DII), Università Politecnica delle Marche, 60131, Ancona, Italy (e-mail: a.bonci@univpm.it, massimiliano.pirani@gmail.com)*

The path towards Industrie 4.0, requires that factory automation problems cope with the cyber-physical system complexity and its challenges. Some practical experiences and literature in the field testify that the role of the database management systems is becoming central for control and automation technology in the new industrial scenario. This article proposes database-centric technology and architectures that seamlessly integrate networking, artificial intelligence and real-time control issues into a unified model of computing. The proposed methodology is also viable for the development of simulation and rapid prototyping tools for smart and advanced industrial automation.

Keywords: Computational methods, Database management systems, Distributed computer control systems, Embedded systems, Industry automation, Intelligent manufacturing systems, Simulators.

1. INTRODUCTION

The factory of the future is in the objectives of the *Industrie 4.0* strategy. Cyber-physical systems (CPSs) are the new research framework that makes the complexity of the Industrie 4.0 goals treatable and where physical and software components are deeply intertwined to interacting each other in a myriad of ways that change with context (Lee 2015). The efficiency measurements and assessment of industrial production processes in the future scenario have to take into account the increasing role of information flow across processes from the enterprise system level down to the shop-floor. Information is the main vehicle that allows humans to be in control of sustainability and productivity and allows the introduction of artificial intelligence as a decision support tool. The appearance of unforeseen behaviours is a typical phenomenon of complexity: new features and prospects might emerge from the deliberate application of data mining techniques coupled with artificial reasoning and inference on already well known and established data. Indeed new opportunities would originate from the full exploitation of information acquired, stored and communicated in industrial processes. Most of it unfortunately, due to the high volume of data, is usually doomed to be neglected as noise rather than useful added-value information. Though expensive smart metering methods are common, on current manufacturing plants, they should deserve a deeper exploitation.

This paper proposes an "information-conservative" approach suggesting a key enabling technology and a methodology for modeling, control, simulation, planning, optimization and scheduling of industrial processes, through dynamical assessment of some common key performance indicators (KPIs). The key approach is the pervasive use of relational database systems that actively support transmission, storage and elaboration of information across the 5 levels defined in the ISA-95 standard – from sensing and actuation to the management of a network of enterprises. The traditional 5 ISA-95 levels seem designated to be blurred and surpassed soon by the new smart factory technologies and concepts. It is expected to move from the existing hierarchical control structures, based on the ISA-95 automation pyramid, towards more decentralized and reconfigurable structures based on the CPS principles (Leitao 2015). Indeed cyber-physical production systems (CPPSs) will show cyber capabilities within every physical component, as distributed computing along with distributed intelligence, and 'self-*' methods, namely: self-adaptation and self-configuration, along with self-diagnosis, self-organization and self-reconfiguration dynamics, as required through the *Industrie 4.0* strategy.

The introduction of an active distributed database mechanism at the shop-floor, through the best embedded database technologies today available, renders the data mining and the optimization of processes viable for the CPS challenge. By the use of the quality of a declarative language, as the database languages in the relational model, most of the techniques for planning and optimization (Jeon 2016) can be enabled dynamically. Decision support systems based on time-aware relational model inference can lead towards results potentially unforeseen at the beginning of the information gathering (Yang 2016, Nickel 2016, Date 2014). The full relational model will require more scientific effort in the future but a restricted database-centric technology based on the established SQL database language standard can create a first technological step towards the challenges made up by the smart manufacturing scenario.

We propose guidelines and technology hints that can be effectively used in KPI-based control for the energy efficiency of industrial processes within the sustainable factory of the future research framework (Stiel 2016).

In section 2 we introduce related work on the basic ideas of the database-centric technology. In section 3 a problem

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.608

description along with a technology description and its unifying role for industry is provided. In section 4 a simulation and modelling methodology is introduced. In section 5 the results expected from the methodology are discussed. Section 6 is for conclusions.

2. PERVASIVE DATABASE TECHNOLOGY AND RELATED WORK

Information is gold, and nothing of it should be lost. On the top of it we can decide different views with their different resolutions and granularity. As the level of aggregation of information increases, the influence of changing structural effects and other factors are lost as some emergence over the data might remain statically frozen by aggregation itself. Going down towards the lower and more detailed levels increases the understanding of the multitude of factors that affect energy efficiency, to make smart production decisions. However, as we go to details the quantity of data and sensing becomes complex, and this complexity is unavoidable in the cyber-physical systems scenario. We have to cope with that complexity in a clever and lean way. The appropriate level of detail for the construction of energy efficiency indicators can be controlled through the application of a methodology based on pervasive database management systems (DBMSs).

Industrial production processes are complex and the heterogeneity of the components and protocols across the factory floor, and its organizational and geographical boundaries, create strong demands for system integration and interoperability (He 2014). Valuable unifying efforts towards standards, like the OPC UA (Mahnke 2009), have been well established. In any case, this computing model might not be completely able to scale down and keep up with the incoming CPS storm, mainly because those issues were not present at the time of its design. While in OPC UA the database is used mainly for historical data storage, in our proposed approach, each component or agent in the process is provided with active database capabilities that tracks data and events to trigger control on the plant. Information is made constantly available for its use at the highest strategic levels, where decisions cope with the long-term issues. Decisions made at the strategic level can influence the procedures implemented at lower levels through dynamical programming based on planning and reasoning with context awareness. By using distributed computing techniques and tools, we can keep the costs of complex solutions low and viable. The technique of adding a database everywhere does not necessarily require a change in the lower level technology or acquisition system. It can be applied on top of existing facilities. Distributed databases for CPSs imply networking as we have to include the Internet of things (IoT) inside. In (Jia 2013) it is shown how the IoT promise of integrating the digital world of the Internet with the physical world needs to be implemented through a systematic approach for integrating the sensors, actuators, where data is the central entity for the realization of context-aware services. In (Rajhans 2014) it is shown how abstraction of models is necessary in the current complexity of CPSs. For this kind of abstract modelling we put forth the expressiveness of the relational model, or where not fully available, at least the ordinary relational database technology with the plethora of extensions provided from SQL database systems vendors. Nowadays, some industrial producers like Inductive Automation™ already use this kind of database-centric technique (white paper 2012). They use a SQL database as the centre of every software and business logic of their industrial automation application. In (De Morais 2014) the author shows how data processing can be integrated and performed within the DBMS. Both the formerly cited solutions are going in the right direction lest the necessary scalability foreseen in cyber-physical systems is still lacking. In CPSs the database must scale down to use a minimum of resources (potentially below 1MB of memory). Therefore, the SQL DBMS implementation needed, must fit the requirements of special purpose embedded solutions. A viable DBMS implementation could be represented from the very portable and mature SQLite which is released in public domain and is apt and open to innovation and research. Nevertheless, a major drawback in the SQL approach is the SQL itself. Notably, SQL falls short when it has to cooperate with artificial intelligence. Although SQL database language is a declarative and 4th generation programming language, it doesn't supply the expressiveness needed, for example, in first order logic inference problems. The real model that should be adopted and implemented soon is the relational model in the original Codd's sense (Codd 1990) reviewed by some followers (Darwen 2012). The relational model in general can act as a higher-order logic representation language, so fit to cope with all problems and inherent limitations of SQL (e.g. SQL has not a standard query for asking a list of the database tables, or treat a table itself as a variable. This is usually overcome by producers with proprietary extensions). Unfortunately, no lightweight implementation of the relational model is already available. Until the availability of research efforts in the relational model implementation sense, we are left with SQL and its shortcomings. Besides, SQL based automation still can represent a great step forward in the paradigm for industrial process control right now. Means and techniques to adapt SQL to artificial intelligence (AI) have been already developed (Schuldt 2008, AlAmri 2012, Bhatia 2013). The use of SQL as a base technology for AI and automation is promising and is the straightest path towards managing the complexity of CPSs challenges.

3. FACTORY AND PROCESS AUTOMATION PROBLEM

A DBMS-centric approach for complex process automation was already looked into through the federation of MySQL and PosgresSQL technologies in 2008 in the developments of a European FP7 research project (CAFE 2008). MySQL was used for embedded units as a moderately lightweight DBMS solution for embedded Remote Terminal Unit (RTU) boards while PostgreSQL was used as the remote global data centre. The major concept put forth there was the unifying role of the DBMS to host heterogeneous technologies for acquisition, actuation and data processing. With suitable and simple adapters, any sensor and actuator, along with special purpose computing units and machines, were connected in a whole unified DBMS informational centre. In Figure 1 we show the unifying role of DBMS-centric technology in an architecture that copes with the problem of the heterogeneity of the automation components.

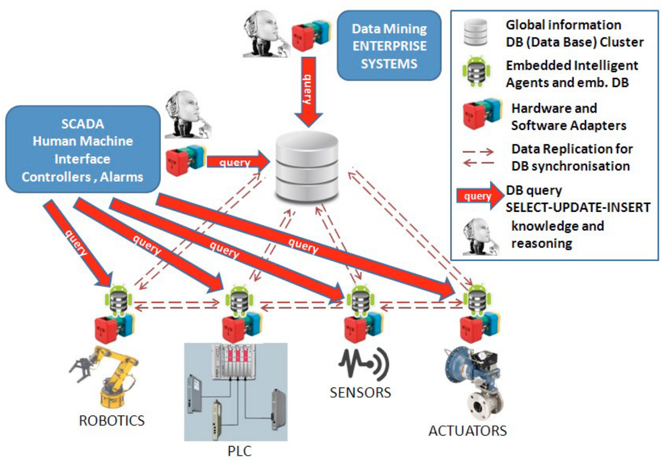

Figure 1: the unifying role of database centric approach.

Note that the DBMS must be a distributed system. The distribution of the information and processing is carried on by a lightweight synchronization mechanisms obtained by a well calibrated data replication. A distributed infrastructure is obtained when suitable mechanisms are used to propagate the updates on database tables across the distributed DBMS units. Therefore, the UPDATE or INSERT on a table, due to a new acquisition is propagated to a networked DBMS that hosts a logic that needs the information update as input.

4. DISTRIBUTED DBMS-CENTRIC METHODS

4.1 Lightweight Database Synchronization through Distributed Replication

Through the replication mechanism, a selective and optimized synchronization among the computing actors is obtained. Through appropriate configuration programming (in database language, of course), the set of subscriber components asynchronously receive updates through the networking infrastructure. Information is propagated and communicated across the whole infrastructure through appropriate triggers registered in the database tables. This allows every device and controller to be in contact through the data simply by querying the database. Each embedded database can synchronize centrally or with a neighbor by means of a lightweight replication mechanism. In SQLite, for example, these kinds of triggers are easily implemented. A trivial example of *trigger* code is provided, with the assumption that table *v1* has *value* and *timestamp* columns (attributes):

```
CREATE TRIGGER vardisplay AFTER INSERT ON v1
  BEGIN
    SELECT publish_value(NEW.value,
                         NEW.timestamp);
  END;
```

The *publish_value()* procedure can be written in C, or in any other language, and might transmit the value to a distant component's database that requires it for its own device control purposes. Some assumptions can be made that allow synchronization efficiency. It can be assumed that a variable *xxx* is unique and has an absolute identifier (at present, the IPv6 addressing scheme appears to be the best identifier choice). The variable *xxx* might be associated to a physical object in the universe. The *xxx* identifier must anyway be unique and reserved forever (like a Web URI). It has to be assumed that the process that affects the values of variable *xxx* is unique, whether it is a mere sensor acquisition task or complex processing. It can further be assumed that *xxx* is a primary variable as it is the only source of the information related to *xxx* in the whole universe, and so it is called a *Master* variable. A *Master* variable is a variable that is physically acquired by a device (e.g., embedded board, general purpose computer, software) or is generated through calculation from one or more acquired variables. *Slave* variables are then defined as those variables that need to be synchronized to a certain *Master* through replication. Of course, a process that affects a *Master* variable can also concurrently be fed as a *Slave* from other *Master* or *Slave* variables. A topological structure is then obtained that allows all of the sources of information to feed all of the distributed components that subscribe to the *Master* variables. Note that the replication configurations are themselves database variables as everything should be a relational variable in the relational model. They can be shared and moved across the network as the processes that use them are also distributed, but unique (similar to a Domain Name System).

4.2 Publish-subscribe paradigm and the Internet of Things

The formerly explained replication technique addresses the major requirement for communication in CPSs. In this context communication is the Internet of things (IoT). It is well known that a challenge posed by the IoT is the search for a unified and lightweight means of communications that allows machine-to-machine (M2M) intelligent connection (Al-Fuqaha 2015). Breaking new ground in pervasive and distributed computing is the *swarmlet* concept. Swarmlets are applications and services that leverage networked sensors and actuators with cloud services and mobile devices. Their architecture is conceived to embrace heterogeneity instead of attempting yet another unreasonable standardization (Latronico 2015). The proposed DBMS replication technique is candidate to be easily compliant with the major architectures in IoT. As described in section 3.2 our proposal follows a publish/subscribe scheme where each RTU or software acts like a human being that only subscribes to interesting events/information and publishes relevant events or information. Prominent examples of publish/subscribe schemes are MQTT (Locke 2010), robot operating system (ROS: http://www.ros.org/). A promising framework is the DDS (Data Distribution Service, at http://portals.omg.org/dds/). DDS addresses data in a manner similar to relational databases and recently has been considered its optimization for the IoT (Beckman 2015). Furthermore, the Allseen Alliance (https://allseenalliance.org/) puts forth the AllJoyn® Framework. The common concept for all of them is the *topic*, where each node subscribes and some other publishes information. In our DBMS-centric proposal the *topic* is simply the database content itself without intermediate language adapters: the unique language is the database language and its queries. All the information in the process is published to actors and any controller and optimizer can be build over the DBMS infrastructure and its semantic abstraction as a plug-in. However, in a different context, in (De Morais 2014) there is a quite detailed explanation of the advantages of such an active database-centric approach. In the distributed system the DBMS must play the role of inter process communication (IPC) means. In (De Morais 2014) the IPC is mentioned but not used in a distributed sense (because there was only one DBMS computing unit). They use the term *resource adapter* to define a computing processing unit that is connected with physical actuators and sensors and which communicates with a centralized DBMS through the IPC extensions available on PostgresSQL – NOTIFY, LISTEN and UNLISTEN commands. In the industrial distributed control system (DCS) context the same *resource adapter* role is performed by the remote terminal units (RTUs). IPC through DBMS connects the RTUs and higher level central units of the DCS system.

4.3 Remote Terminal Units and plug-ins

Typically, an RTU in industrial control systems is a *holonic* unit which features some local procedural capabilities along with concurrent communication tasks. It is responsible both for real-time autonomous controlling actions and decisions and for communications with higher intelligence layers up to the enterprise management level. Holonic systems are relevant in enabling the multi agent system technology for intelligent distributed control of industrial plants (Leitao 2013). For our purposes, an RTU device features an embedded DBMS and replication. We need to identify and separate 4 parts in such an RTU structure: the input, the database tables, the logical and procedural part and the output. In figure 2 we identify the four holonic RTU parts. The input part contains the hardware and software that converts input information to an INSERT/UPDATE database query. DBMS tables and related triggers constitute the second part. The triggers launch the logic and algorithms relevant to the former event (with a local publish/subscribe scheme as well) on the third part. That might produce a new INSERT/UPDATE query as a result or a physical output to be handed off to the output part. Concurrently, information is published or received with the replication mechanism across the whole DCS. When a sensor acquires new data it is put in the DBMS through an INSERT command. The new record is transmitted to other holons or to a central DBMS by simply transmitting the same database query through the network. If some distributed business logic depends on that information, a database trigger calls the associated procedure. The output can in turn be propagated to other units. The same mechanism, in reverse order, is used for actuators. In this computing infrastructure the database tables are the globally shared storage part with their triggers for IPC.

The components see the database tables as the unified and unique information interface available. With such an infrastructure, there is fertile land for intelligence and logic to grow and develop.

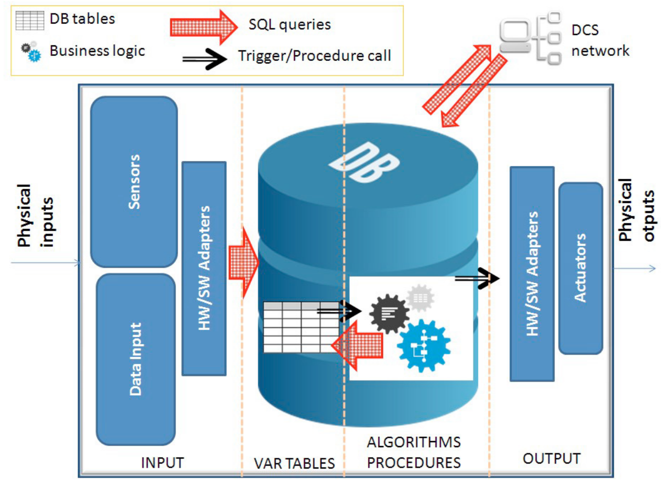

Figure 2: significant functional parts of the RTU

With the infrastructure available, algorithms can easily be plugged into the physical world as a back-end while they see a unified database front-end that enables controllers and other software to be seamlessly interconnected. Every algorithm developed is deployed as a *plug-in* of the system. Every *plug-in* can be immediately put into communication with the others. This enables a vast class of control topologies and hierarchies and any controller software is a *plug-in*. With the plug-in concept, an abstraction layer has been created that enables scientists and engineers to operate the control framework without being concerned about the low-level communication details. A *plug-in* can be uploaded, enrolled and started by a web-based graphical interface.

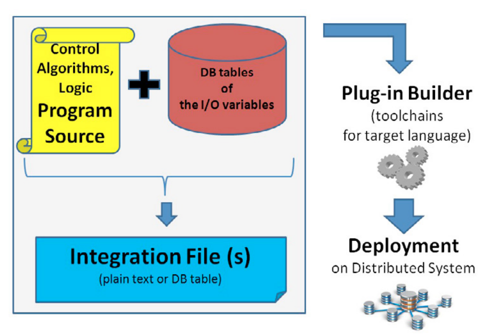

Figure 3: the plug-in design and deployment workflow.

Figure 3 illustrates the workflow of *plug-in* development and deployment. What scientists or practitioners need are the knowledge and the semantics of the database variables, along with the controller source code. The controller source code can come from a simulated and calibrated Simulink® or Stateflow® model or any Matlab® or high level programming language. Having in mind the input and output variables, the designer can edit the *Integration File* with the assignment of the inputs/outputs of the control algorithm to the corresponding database variables – the Integration File as a set of relations must be encoded as a database table as well. The source code is then transformed into the procedures that are triggered by the DBMS to perform the computations.

5 EXPECTED EXPERIMENTAL RESULTS

5.1 Rapid process control prototyping through simulation and modeling

In (Bonci 2014) the authors show how the former *plug-in* concept promotes openness in the rapid prototyping of scientific software, and how it breaks the entry barriers for small and medium enterprises in industrial applications. Through the database abstraction we can model the industrial process automation in two classes of components. The first class is constituted by the database relations (e.g. tables) and communication means (e.g. IPC and the signaling by triggers). The second class is represented by the procedures that implement the controlling algorithms relying on data inputs and outputs, formerly defined as plug-ins. Within the second class, we can also design all the components that constitute a complete model of the physical processes involved in the production. In turn the plug-ins class is to be divided in two subclasses, one for the control and one for the models suitable for simulations and model-based control (e.g. in model predictive control techniques).

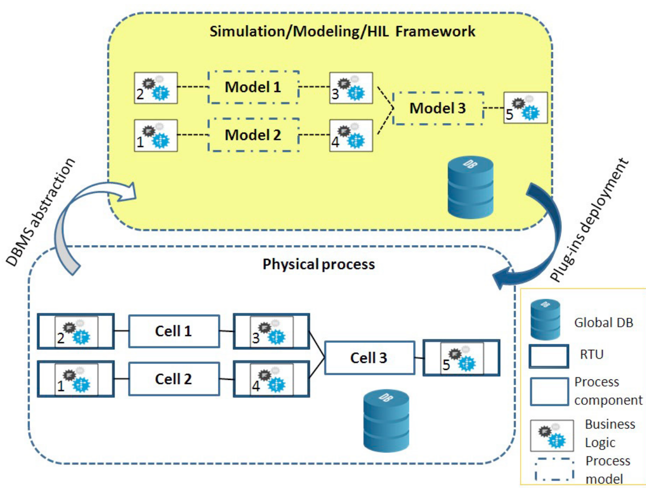

Figure 4: DBMS abstraction and development framework

In Figure 4, we try to represent how the DBMS-centric infrastructure and related plug-ins are a useful scheme for the creation of tools for the rapid prototyping and simulation of a production plant. Indeed, by switching between simulation and physical acquisition we can create models of the plant or calibrate the controllers as well. Furthermore, by abstracting the business logic blocks and the physical plant blocks over the ubiquitous database infrastructure, we can functionally replace them with the plug-ins that model the control and business logic and the physical plant models in a virtual environment. These blocks can be simulated, tested and calibrated in a suitable development framework and then deployed back into the physical RTUs. Figure 4 expresses the concept of the database as the fundamental and invariant ground enabling the architectural link between the physical and the virtual industrial plant. The dotted lines in the modelling framework are not real connections but associations with the suitable variable tables. They are transformed back into physical communications through the deployment process. The implementation of algorithms through stored procedures or triggers, can be challenging. A best practice suggested here is to separate clearly the data infrastructure from the business logic. Once the data input and outputs are available globally through the DBMS relations (tables) we can abstract and simulate the business logic that constitutes the algorithms that produces outputs from the inputs, develop them with preferred language, test and calibrate and then compile them back into the DBMS-centric implementation through the plug-ins approach.

5.2 The Virtual framework development

The choice of the virtual framework is quite immaterial as far as the components of the simulation can be connected to a DBMS. Suitable APIs exist for the majority of programming languages. Matlab Simulink® and Stateflow® are quite a de facto standard for researchers in the control community. Future work will focus on these tools for a validation of the proposed methodology. The tools will allow these three phases to be performed easily:

1. Creating models of the production lines, environment and energy consuming components.

2. Gathering the real data and calibrate the models through real incoming data.

3. Using the output of the calibrated models to assess the KPI effectiveness and optmimize through data-mining on the raw not aggregated data. Optimized blockset is implemented back as a plug-in on the real plant.

The actual novelty, is on paying due attention to the capabilities of embedded DBMS for the dynamical modeling and operation of hybrid systems like the production processes.

A good starting point for the next experimental work is the set of already available blocksets for the Simulink environment that support the publish/subscribe scheme and already developed for: ROS (Matlab Robotics System Toolbox®); DDS (MathWorks provides Simulink® blocks and MATLAB® classes for RTI Connext DDS); and in the RT-LAB Orchestra® product (http://www.opal-rt.com). The design of the tools starts with the connection of every blockset to a publisher (outputs) and a subscriber (inputs) blockset.

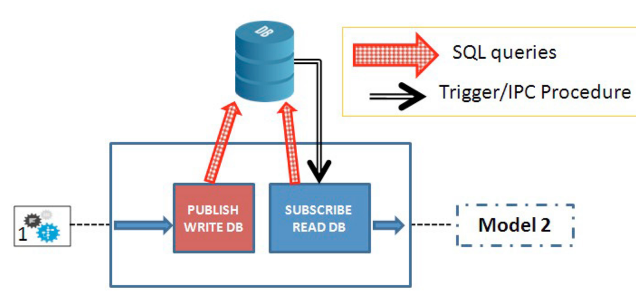

Figure 5: desired functionality of the new Simulink signal/bus element (central box)

The publish/subscribe scheme can be achieved by forging a tailored new signal/bus object in Simulink that transforms the blockset outputs into database INSERT/UPDATE queries.

On query execution, an IPC event triggers a SELECT query for the retrieving of the blockset input values. This way, the simulator tool is controlled by the DBMS. Note that this also allows hybrid virtual configurations like the hardware–in-the-loop techniques (see for example the VxWorks™ *Async Interrupt* and *Async Task* blocksets within the Real Time Workshop Toolbox®). Figure 5 shows the functional scheme of the signal object component under development.

5. CONCLUSIONS

This paper proposes a viable enabling technology for the harnessing of complex issues in the incoming scenario of industrial automation where the smart features of cyber-physical systems are going to play a leading role. Based on lessons learned, state of the art in recent literature and commercial trends, a database-centric architecture is proposed as a suitable solution for the problems in networking and control of factory and process automation of the future. Moreover, the proposed technique allows the development of simulation, modelling and rapid prototyping tools by leveraging the best available standard approaches and technologies. Future work suggested is the development of scientific research for the enforcement of the full relational model with efficient practical implementations. In the short term a set of new Simulink tools are going to be developed for the expected development framework along with tests of advanced control techniques for KPI-based optimization of production and energy efficiency of real processes.

REFERENCES

AlAmri, A. (2012). The relational database layout to store ontology knowledge base. *Information Retrieval & Knowledge Man. (CAMP), 2012 Int. Conf. on,* pp.74,81.

Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *Communications Surveys & Tutorials, IEEE*.

Beckman K., Dedi O. (2015), sDDS: a portable data distribution service implementation for WSN and IoT platforms. *IEEE WISES 2015, Proc. Of 12th Int. Workshop on Intelligent Solutions in Embedded Systems, Oct. 29-30, 2015, pp. 115-120*

Bhatia, P., Khurana, N., Sharma, N. (2013). Intuitive Approach to Use Intelligent Database for Prediction. *Int. J. of Computer Applications,* vol. 83, no. 15, pp. 36,40.

Bonci, A., Imbrescia, S., Pirani, M. Ratini, P. (2014), Rapid prototyping of open source ordinary differential equations solver in distributed embedded control application. *Mechatronic and Embedded Systems and Applications IEEE/ASME 10th Int. Conf. on , pp. 1-6*.

CAFE project, EU FP7-KBBE, g.a. 212754 (2008). Deliverable. D.8.6, "The integrated system and the software main frame".

Codd, E. F. (1990). *The Relational Model for Database Management: Version 2*. Addison-Wesley Longman Publishing Co., Boston, MA, USA.

Darwen, H. (2012). *An Introduction to Relational Database Theory*, Ventus, 2012. Available: *http://bookboon.com*).

Date CJ, Darwen H, Lorentzos N. (2014) *Time and relational theory: temporal databases in the relational model and SQL*. Morgan Kaufmann.

De Morais, W. O., Lundstrom, J., Wickstrom, N. (2014). Active in-database processing to support ambient assisted living system. *Sensors.,* vol. 14, no.8, pp.14 785.

He, W., Xu, L. D. (2014). Integration of Distributed Enterprise Applications: A Survey. *IEEE Trans. Ind. Informat.,* vol.10, no.1, pp.35,42.

Inductive Automation white paper (2012). *SQL: The Next Big Thing in SCADA, How SQL is Redefining SCADA.* Available: *www.inductiveautomation.com*

Jeon, S.M., Gitae K. (2016) A survey of simulation modeling techniques in production planning and control (PPC). *Production Planning & Control*, pp. 1,18.

Jia, Z., Iannucci, B. Hennessy, M., Xiao S., Kumar, S., Pfeffer, D., Aljedia, B., Ren, Y., Griss, M., Rosenberg, S., Cao, J. Rowe, A. (2013). Sensor Data as a Service -- A Federated Platform for Mobile Data-centric Service Development and Sharing. *Services Computing (SCC), 2013 IEEE Int. Conf. on,* pp.446,453.

Latronico, E.; Lee, E.A.; Lohstroh, M.; Shaver, C.; Wasicek, A.; Weber, M. (2015). A Vision of Swarmlets. *Internet Computing, IEEE , vol.19, no.2, pp.20,28*.

Lee, E. A. (2015). The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors,* vol 15, no. 3:pp. 4837-4869, 2015

Leitao, P., Marik,V., Vrba, P. (2013). Past, Present, and Future of Industrial Agent Applications. *Industrial Informatics, IEEE Trans.*, vol.9, no.4, pp.2360,2372

Leitao, P., Barbosa J., Papadopoulou, M.-E.C., Venieris, I. S. (2015). Standardization in cyber-physical systems: The ARUM case. *Industrial Tech. (ICIT), 2015 IEEE Int. Conf.,* pp. 2988,2993, Mar 2015.

Locke, D. (2010). Mq telemetry transport v3. 1 protocol, specification. *IBM developerWorks Technical Library*

Mahnke, W., Leitner, S., Damm, M. (2009). *OPC Unified Architecture.*

Nickel M., Murphy K., Tresp V., Gabrilovich E. (2016). A Review of Relational Machine Learning for Knowledge Graphs. *Proc. of the IEEE, vol.104, no.1, pp. 11,33*.

Rajhans, A., Bhave, A.; Ruchkin, I.;Krogh, B.H.;Garlan, D.; Platzer,A.;Schmerl, B. (2014). Supporting Heterogeneity in Cyber-Physical Systems Architectures. *Automatic Control, IEEE Trans. on,* vol.59, no.12, pp.3178,3193

Schuldt, H. (2008). Agent and Databases: A Symbiosis ?. *Cooperative Information Agents XII, 12th International Workshop, September 10-12, CIA 2008, Prague*.

Stiel F., Michel T., Teuteberg F.. Enhancing manufacturing and transportation decision support systems with LCA add-ins. *J. of Cleaner Production*, vol. 110, pp.85,89

Yang, S., Khot T, Kersting K., Natarajan S. (2016). Learning Continuous-Time Bayesian Networks in Relational Domains: A Non-Parametric Approach. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.