Development of a software framework for real-time management of intelligent devices

T. Naumović, ORCID: 0000-0001-9849-7665 <tamara@elab.rs>
L. Baljak, ORCID: 0000-0003-3779-7335 <lukabaljak@elab.rs>
L. Živojinović, ORCID: 0000-0003-5356-3146 <lazar@elab.rs>
F. Filipović, ORCID: 0000-0001-7113-3802 <filipfilipovic@elab.rs>
Faculty of organizational sciences, University of Belgrade,
Jove Ilića 154, 11000 Belgrade, Serbia

Abstract. The subject of this paper is development of software framework for real-time management of intelligent devices. The framework enables intelligent management of IoT devices in cyber-physical systems using models based on recurrence relations and differential equations. The platform was developed using Python programming language, Django framework and wide corpus of modules and libraries that supports continuous simulation. The software framework incorporates application programming interface as well, for specification of system behaviour, transmission of input parameters and output results, sending control actions via web services to the IoT system.

Keywords: software framework; IoT; continuous simulation; python


Acknowledgments. Authors would like to thank prof. dr. Bozidar Radenkovic, prof. dr. Marijana Despotovic-Zrakic, prof. dr. Zorica Bogdanovic, prof. dr. Dusan Barac and prof. dr. Aleksandra Labus for guidance and mentoring throughout this research and software development.

1. Introduction

Cyber-physical systems (CPS for short) integrate devices, networks, interfaces, computer systems, and others with physical world. The fact that those elements are heterogeneous, hybrid, distributed and numerous, makes their analysis, design and implementation quite challenging and complex. In addition, CPSs are real time by their nature. Wide corpus of services, applications and interactions within CPS as well as huge growth of Internet of things further fuelled the need to change and improve existing approaches to managing those systems [1][2]. One of the most significant issues is to explore and model properties of CPS elements, their connections and behaviour [3][4]. CPS immerged from the integration of devices with embedded systems, smart objects, people and physical environment typically connected via communication structure. Thus, it is no surprise that smart environments and systems are among the fields of CPS application.

Smart systems are integral part of CPS. The key technology for developing cyber physical systems is Internet of Things, IoT [5][6][7]. According to [8], cyber-physical systems, Internet of things and big data are related concepts of cooperative solutions, where people, autonomous devices and the environment interact with one another to achieve a certain goal. IoT technologies enable the connection of a large number of users, devices, services and applications to the Internet [9]. Management of intelligent devices often needs to be done in real-time. Real-time Control System (RCS) is a reference model of architecture that defines the types of functions needed for intelligent real-time control [10]. RCS provides a comprehensive and basic methodology for design, engineering, integration and testing of control systems [11].

In RCS systems, the state of many variables changes continuously over time, so the management of these systems can be modelled using differential equations and recurrence relations. Hence, simulation enables investigation of behaviour of such dynamic systems by developing appropriate models and using these models in experiments designed to provide an insight into the future behaviour of the system under specific conditions [12][13][14]. Simulation of CPS is becoming extremely important for both academia and practice as results of simulations have huge potential to be applied in research, business and engineering [15].

The main idea of the research is to develop a comprehensive platform that would enable modelling and simulation of different smart environments. To achieve this goal, it is vital to define a uniform formal model applicable to any smart environment whose mathematical representation can be mapped to its implementation as one-to-one correspondence. The software framework for real-time management of intelligent devices represents a cyber-physical system incorporating IoT devices as the physical component of the system and software framework accompanied with required network infrastructure as the cyber component. Having available information and input data from intelligent devices in real-time allows the simulation engine, as an integrated part of the solution, to calculate and create a plausible outcome. On the other hand, outcome created as the result of the simulation enables investigation of behaviour of such dynamic systems by developing appropriate models and using these models in experiments designed to provide an insight into the future behaviour of the system under specific conditions [12][13][14]. Simulation of CPS is becoming extremely important for both academia and practice as results of simulations have huge potential to be applied in research, business and engineering [15].

The main idea of the research is to develop a comprehensive platform that would enable modelling and simulation of different smart environments. To achieve this goal, it is vital to define a uniform formal model applicable to any smart environment whose mathematical representation can be mapped to its implementation as one-to-one correspondence. The software framework for real-time management of intelligent devices represents a cyber-physical system incorporating IoT devices as the physical component of the system and software framework accompanied with required network infrastructure as the cyber component. Having available information and input data from intelligent devices in real-time allows the simulation engine, as an integrated part of the solution, to calculate and create a plausible outcome. On the other hand, outcome created as the result of the simulation can be a trigger dispatching control actions towards the IoT system. The formal model, implementation and example illustrating the applicability of the presented mathematical model will be explained further in the paper.
2. Formal Model

2.1 Continuous system simulation in IoT context

Continuous system simulation refers to experimenting with models whose states are changing continuously in time [11]. These types of simulation systems are often described by differential equitation. Time is independent variable in most cases. Continuous simulation can be used in different contexts and covers numerous types of real-world problems [16]. Considering time as an independent variable, digital computer has constraints solving differential equations, which is why it was necessary to develop a specific language to resolve this issue.

Different specialized languages for continuous simulation were developed, such as: CSMP (Continuous Simulation Modelling Programme) ESL (European Simulation Language) ACSL (Advanced Continuous Simulation Language) CSSL4 (Continuous System Simulation Language), Simulink, Matlab, Modelica and others that have been developed to simplify modelling, and to minimize problems related to programming continuous systems [16]. However, a majority of simulation tools have limitations related to low level of flexibility and adaptability, high costs, platform dependence, maintenance difficulties, etc. [16]

CSMP/FON platform for continuous system simulation was developed following these principles [17]:

- minimize required hardware resources and improve speed of execution;
- suitable and easy to use for educational purpose;
- simple and rich user interface;
- support for scientific research;
- saving costs.

The CSMP/FON is an open source solution and can be downloaded from the web site https://elab.fon.bg.ac.rs/softver/csmmp. It has been used for many years in research and teaching within simulation related courses at University of Belgrade.

Software framework for real–time management of intelligent devices and IoT systems in general is a time dependent system, which requires a tool that can overcome any time – related performance issues. Ergo, using CSMP simulation logic in the software development process was a way of introducing control mechanism in the system.

2.2 Formal model of a continuous simulation system

Formal model of a continuous simulation system can be given as a tuple [18]:

\[ M = (U, Y, S, \delta, \lambda, S_0) \]  \hspace{1cm} (1)

with the following meanings:

- \( U \) – set of inputs;
- \( Y \) – set of outputs;
- \( S \) – set of state variables;
- \( \delta \) – transfer function: \( \delta: U \times S \rightarrow S \);
- \( \lambda \) – output function: \( \lambda: U \times S \rightarrow Y \);
- \( S_0 \) – set of initial states.

Function of a variable \( \varphi \) is a mapping of a non-empty set \( X \), of variables \( x \), signed as domain, in non-empty set \( Y \) of variables \( y \), signed as scope (or codomain, set of function values) [18]:

\[ \varphi: X \rightarrow Y; \]

a function of many variables is presented through mapping:

\[ \varphi: X \times X \times X \times \ldots \times X \rightarrow Y; \]

a block is presented as ordered set of three elements

\[ b = (\phi, X, Y), \]

each \( x \in X \) is input, while \( y \in Y \) is output from the block.

The process of continuous simulation is based on solving differential equations and recurrence relations [17][18]. CSMP simulation language is block-oriented languages designed for solving systems of differential equations. Each block is specified by a set of inputs and parameters and a graphic symbol [12]. The graphic display of elements in the general form is presented in fig. 1.

2.3 Formal model of a hybrid IoT system for real-time simulation

The current simulation model describes a system that allows solving differential equation systems in the given time with predefined variables and inputs [17][18]. The software framework for real-time management of intelligent devices requires a broaden model that will be suitable for use in real–time IoT systems [19].

Fig. 2 presents the concept of a hybrid IoT system for real-time simulation. This model enables having values measured in the environment in real-time included as variables of the simulation systems. In addition, the model enables managing the IoT system using variables obtained through the simulation.

![Fig 2. Hybrid IoT system for real-time simulation](image)
The operation $\rho$ is the operation of reading the values of variables from IoT system. These values can be used in the simulation system for calculations. The operation $\omega$ enables writing the values of variables into the IoT system. These values are calculated in the simulation engine and then sent to the IoT system. These values can also be used for triggering specific actions of the IoT system. The operation $\gamma$ enables reading the outputs of the IoT systems.

Having in mind that IoT systems are distributed, all these operations for interaction between the simulation and IoT systems need to be realized via web, using web services. Depending on the scenario, both PUSH and POP methods can be used.

After extending the formal model of continuous simulation system with the IoT elements, the process of continuous system simulation can be described with the finite automata equations [18]:

$$S(t) = I \times A_1 \cdot (U(t), S(t)); \quad Y(t) = A_2 \cdot (U(t), S(t))$$

where $A_1$ and $A_2$ are matrix representations of algebraic functions, and $I$ is the matrix representation of the integration operator (fig 3).

![Fig 3. The structure of finite automata for simulation of continuous systems [18]](image)

A more granular structure of continuous system simulation is presented in the fig 4.

Fig. 4 depicts the decomposition of the operator $A_1$ to its elementary and primitive functions, represented as algebraic blocks. As explained later in section D, input of every block is an element that can come from either a set of inputs, a set of state variables or a set of the associated variables that represent inputs of the preceding algebraic blocks.

### 2.4 Orderliness

The essential feature of any non-trivial mathematical model of the continuous simulation is the feedback. The feedback occurs in the model as a result of a chain of cause-and-effect that generates a loop [20]. Considering the case of continuous simulation the model develops the feedback loop if it is impossible to mark all blocks from the set that satisfy a condition $i < j$, where block’s $b_i$ output is connected to block’s $b_j$ input [17]. The feedback loop imposes a compulsory requirement for computability of mathematical model called orderliness, defined as:

The set $A$ of all countable algebraic blocks (blocks that correspond to algebraic functions) of $M$ models is called orderly if all distinct objects $a_j \in A$ can be ordered (sorted) in such linear list where inputs of every distinct object $a_j$ are elements of some of the following sets [18]:

1. $U$ – set of inputs;
2. $S$ – set of state variables;
3. Subset $C \subset \mathcal{C}$ defined as: $\mathcal{C} = \{c \in \mathcal{C} | \forall i < j, \forall a_j = (\varphi_i, X_i, Y_i) ; c \in X_i\}$.

### 3. Mapping mathematical model to implementation

Mapping the mathematical method given in equations 1-9 is represented through series of UML sequence diagrams, where each method has its corresponding diagram.

The implementation of the software framework described through this research will be based on the concurrent computing and NoSQL concepts, such as threads and use of the MongoDB document-oriented database program.

![Fig 4. Block diagram of granular structure of continuous systems](image)
embed them as a part of the continuous simulation process, it will send a control action to the IoT system as a result of the continuous simulation process or it will read the output from the IoT system (fig 6).

Fig 5. UML sequence diagram of the core simulation process

Fig 6. UML sequence diagram of processing calls to IoT system
3.2 Mapping the values of state variables and output from the IoT system

The call for executing operation $\rho$ (5) – reading the values of variables from IoT system, is illustrated in the fig. 5 as a part of the calculation process, where simulation engine should consider IoT values as part of the calculation.

The control actions, $\omega$ (6) – writing the values of variables into IoT system, sent to the IoT system are, also, a type of call. By connecting to the IoT system, the engine can access its API, create a call to the function provided by the user, send data from the simulation engine and thus begin the given process on the IoT platform. Such call is illustrated in the fig. 7.

Through the connection made to IoT system our engine can retrieve IoT system outputs and display them through the platform interface, which is directly correlated with mathematical operation $\gamma$ (7) – reading the outputs of the IoT systems (fig 8).

3.3 Example: Smart watering system simulation

The example of smart watering system simulation is an illustration of the operation $\omega$ (6), where control actions and variables are being sent to IoT system. For this example it is necessary to create control actions that will forward the data collected through the simulation of the environment and air humidity by the simulation engine, and signal the beginning of the IoT system actions.

Smart watering system is based on air humidity predictions, provided as input parameters given by the simulation engine. If the humidity is under the marginal value set in the IoT system, the watering process begins.

4. Conclusion

Modelling hybrid IoT system for real – time simulation presents a focal point of this research. Thus, successfully mapping the values of state variables from the IoT system in the implementation process is essential.

The autonomous performance of the simulation program should be implemented using the concepts of concurrent computing – threads:
1) servicing requests for the simulation process control and error reporting,
2) servicing requests for configuration changes,
3) reading data and sending control actions to IoT system,
4) servicing requests for simulation results,
5) execution of the simulation process

Further research and work should be directed towards execution of the proposed implementation, integration of the platform in the students’ educational process and evaluation and revision of the software performance. Upgrading the existing model with new modules should be considered.

References

Наумович Т., Балjak Л., Живоинович Л., Филипович Ф. Разработка программной среды для управления интеллектуальными устройствами в реальном времени. Труды ИСП РАН, том 31, вып. 3, 2019 г., стр. 35-46


Информация об авторах / Information about authors

Тамара НАУМОВИЧ получила степень магистра в 2018 году и в настоящее время готовит диссертацию на кафедре электронного бизнеса факультета организационных наук Белградского университета, Сербия. Ее область интересов включает Интернет вещей, физическое и непрерывное моделирование систем, а также разработку программного обеспечения.

Тамара NAUMOVIĆ received the M.S. degree in 2018 and is currently pursuing the Ph.D. degree at the Department of e-business Faculty of Organizational Sciences, University of Belgrade, Serbia. Her research and field of interest include Internet of Things, physics and continuous system simulation, and software development.

Лука БАЛЯК – студент факультета организационных наук. Его исследовательские интересы включают Интернет вещей, краудсенинг и мобильную разработку.

Luka BALJAK is a graduate student at the Faculty of Organizational Sciences. His research interest include Internet of Things, crowdsensing and mobile development.

Лазар ЖИВОИНОВИЧ – аспирант кафедры электронного бизнеса факультета организационных наук Белградского университета, Сербия. Его исследовательские интересы включают в себя языки моделирования и симуляции, агентное моделирование и электронное обучение.

Лазар ŽIVOJINOVIĆ is a Ph.D. student at the Department of e-business Faculty of Organizational Sciences, University of Belgrade, Serbia. His research interests include simulation and simulation languages, agent-based simulation and e-learning.