A Pattern-Based Approach for Designing Reliable Cyber-Physical Systems

Nikolaos E. Petroulakis[1,2], George Spanoudakis[2], Ioannis G. Askoxylakis[1], Andreas Miaoudakis[1] and Apostolos Traganitis[1]
[1] Institute of Computer Science, Foundation for Research and Technology - Hellas, Heraklion, Greece
[2] School of Informatics, City University London, London
Email: npetro@ics.forth.gr, g.e.spanoudakis@city.ac.uk, asko@ics.forth.gr, miaoudak@ics.forth.gr, tragani@ics.forth.gr

***Abstract*—Cyber-Physical Systems (CPS) appear to be of paramount importance due to their increasing use on critical infrastructure. New challenges have occurred because of the nature and the complexity of such systems in supporting heterogeneous physical and cyber components simultaneously. Failures or attacks on system components decrease system reliability creating severe consequences to CPS and the attached applications. The construction of complex CPS with respect to security and dependability (S&D) properties is necessary to avoid system vulnerabilities at design level. Design patterns are solutions for reusable designs and interactions of objects. In this work we present a pattern-based language for designing CPS able to guarantee S&D properties. The first set of S&D patterns includes the Reliability Component Composition (RCC) Patterns for designing reliable CPS. RCC patterns are encoded in Drools, which is a rule-based reasoning system. To evaluate our approach, we use RCC patterns as a methodology for designing a reliable wireless sensor network attached to a physical architecture to send monitored data to a central controller through relay nodes and paths.**

***Index Terms*—Cyber-Physical Systems; Wireless Sensor Networks; Design Patterns; Security and Dependability; Reliability; Drools.**

I. Introduction

Critical infrastructures such as water and gas distribution networks, power grids, etc. constitute complex Cyber-Physical Systems (CPS) which reliable, safe and secure operation is of paramount importance for daily activities at both national and international level. Government reports such as [1] indicate the growing threats and the clear danger of escalation in the number and the level of sophistication of cyber-physical attacks on CPS which comprise the nerve system of critical infrastructures. For instance, in critical sectors of urban and rural areas such as water distribution and irrigation networks, the quantity of water loss corresponds to more than 50% of the total irrigated water. The existence of old, obsolescent and unprotected systems, malfunctioning and broken pipelines, the danger of physical attacks, water steal and sabotage may be the cause of such losses. Furthermore, cyber-attacks on remote monitored and managed CPS add new security and safety threats, and consequently energy and economic losses.

The term of CPS is used to describe integrations of computation, networking and physical processes [2]. CPS consist of the physical and the cyber part which are able to handle physical and cyber quantities respectively. The physical part of a CPS includes components such as physical plant, sensors and actuators. The cyber part contains communication and process components. The main characteristics of CPS are: (a) the type of process which is needed to be monitored or controlled and (b) the network of intelligent devices that interconnect with a control system. CPS are currently used in electric power generation, transmission and distribution, monitoring systems, water systems, environments, manufacturing, traffic signals, and mass transport [3]. CPS design encounters difficulties because of the nature of the systems, which are time-critical, embedded, fault tolerant, distributed, intelligent and heterogeneous. Research related to CPS focuses on specification analysis, design, verification, and validation of systems that include hardware, software, data, personnel, procedures, and facilities. At design level, the validation and the verification methods for developing Secure and Dependable (S&D) CPS constitutes a critical procedure. The concept of security and dependability should be established on the design phase of such systems.

The purpose of this work is the development of a pattern-based approach for the design of CPS. The main contribution of the approach is that encodes designs of CPS, which are proven to satisfy S&D properties, as CPS design patterns. These patterns can then be used by developers of CPS systems, not necessarily experts in dependability and security, in order to: (a) create designs of their systems in ways guaranteed to satisfy S&D properties, (b) verify if existing designs of CPS systems satisfy required S&D properties and (c) adapt CPS systems at runtime, if necessary, by replacing components in ways that are consistent with and driven by the patterns and, therefore, are guaranteed to satisfy given S&D properties. Based on this approach, we define a first set of CPS patterns with respect to dependability, such as reliability, as a critical attribute of S&D for designing reliable system architectures and networks. Reliability Component Composition (RCC) patterns are expressed in Drools, an engine that enables reasoning driven by production rules [4] which are usually used for business management, software development and service oriented architectures but they can be also applied adequately to the design of CPS architectures.

The remainder of this paper is organized as follows. In section II an overview of related work is presented. In section III, we give a reliability analysis of component composition

on system architectures. In section IV we introduce RCC patterns, a pattern-based approach for validating and designing reliable CPS. In section V, based on a described scenario, we evaluate RCC patterns for the design of monitoring CPS network architectures. Finally, section VI provides conclusions and future steps of our work.

II. Related Work

The design of complex CPS has intrigued the research community and industry [5], [6]. Model-Driven Engineering (MDE) [7] is used as an approach to design secure and dependable CPS [8]. Driven from software development methodology, MDE technologies combine domain-specific modeling languages such as GMF/EMF, UML/SysML or Modelica that can be used to analyze certain aspects of models and synthesize various types of artifacts such as source codes, simulation inputs XML etc. The concept of component-based architecture composition is mainly applied on software components and service oriented architecture but it can be used successfully for designing CPS [9], [10]. Each component can be represented as a service and their correlation can be orchestrated by an orchestration engine. The importance of a semantic model for an effective orchestration of software and physical processes for the design of CPS is pointed out by [2]. Formal approaches and semantics for component-based modeling in CPS are analysed in [11].

Design Patterns are solutions for reusable designs and interactions of objects [12]. Security and dependability patterns are described in [13]. Even though patterns have been mainly applied on software development, the method can be used satisfactorily for the component composition of systems such as CPS [14]. Security patterns, for service compositions based on enabling reasoning engines such as Drools, are described in [15], [16]. In our approach instead of service composition, the concept of component compositions is proposed. Safety and reliability patterns are presented in [17]. The author describes a set of reliable patterns able to offer redundancy on data transmissions for real-time and embedded systems. Reliability appears to be of great importance for the design and the operation of complex CPS [18]. Approaches for designing reliable systems have been presented in works such as [19], [20]. Authors in [21] present formalized architectural patterns for designing reliable CPS that combine fault tolerant architectures with formal verification by developing model compositions in AADL. Reliability estimation through patterns is presented in [22]. The work provides a similar to our approach for estimating the reliability of web service components based on the workflow patterns [23]. However, in our patterns further to reliability estimation, the required reliability is also guaranteed.

III. Component Compositions Analysis

In order to design CPS architectures with respect to S&D, a model-based approach can be used adequately. The representation of CPS as a constitution of physical and cyber component compositions and flows is essential for the model-based design. Complex CPS can be defined as an integration of flows and components. As flows we may consider either the transport of physical quantities or the transmission of computed data. Moreover, the physical and cyber parts of CPS can represent the components of the system. Component-based engineering can be applied for the composition of physical and cyber subsystems of a CPS including also the connectivity between the different components. Flows on physical and cyber networks can be described as a composition of sequence, parallel-split, multi-choice and multi-merge workflows. S&D analysis of CPS includes whether attributes such as confidentiality, integrity, availability, reliability, safety and maintainability are preserved. The conditions depend on the respective S&D property which CPS guarantee.

More precisely for networks, we may consider a network as a component, and the composition of networks as a components composition. The devices included in a network such as gateways, wireless sensors can be defined as system components. But for a wireless link the following can be assumed: either a communication link can be characterized as a component having specific properties (propagation, length, interference, noise etc.) or a link can be a connector which connects two components ie. two wireless sensors. The S&D properties are mainly related to the components which are included in this network. Since we cannot modify easily the medium such as a wireless link, in order to guarantee a security property of the system, the property should be satisfied at the output of the source and at the input of the sink. If those conditions are satisfied, the property shall be also guaranteed at the communication link. In the following sections, we present our approach based on a specific dependability property such as reliability which appears to be of great importance for the design and function of CPS.

A. Component Composition

The main idea is that the composition of subsystems can also compose new systems enhancing their inputs, outputs, properties and attributes. The composition of two atomic components C_1 and C_2 can be defined as $C = C_1 \circ C_2$ having as activity A the composition of activities $A_1 \circ A_2$. On the other hand, we may substitute a system C' with a composition of two objects C'_1 and C'_2 which will invoke and save actions. The generic substitution approach is that components can be replaced by compositions able to perform the same actions. This is mainly related to the type of component composition and flows, and it can be described adequately by workflows of process executions patterns [23]. More specifically, the sequence pattern defines that a process is enabled after the completion of a previous one. The pattern appears as the fundamental approach for building process blocks. The multi-choice pattern (OR-split) provides the execution of a process to be diverged into two or more branches. The parallel-split pattern (AND-split) processes and allows the parallel split into two or more branches. Finally, the multi-merge pattern merges distinct branches into a single branch. In Figure 1, the different

types of component compositions are depicted as workflow patterns.

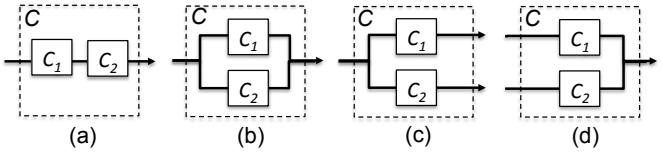

Fig. 1. Component composition as workflow patterns: (a) sequence (b) multi-choice (c) parallel-split (d) multi-merge

B. Validation of Reliability in Compositions

The composition of two components which preserve a S&D property does not necessarily guarantee that the composition will also preserve the same property. In addition, if a composition guarantees the conditions of a S&D property, the atomic components may not preserve the property. In order to validate a S&D property, the reliability property is chosen as a critical condition for the design of complex CPS. Reliability is the ability of a system to perform a required function under stated conditions for a specified period of time [24]. It is an attribute of system dependability and it is also correlated with availability [25]. For hardware components, the property is usually provided by the manufacturer. This is calculated based on the complexity and the age of the component. A very important part of the component composition is the connectivity between the different components and it can be either wired or wireless. Reliability in networks is the probability of successful packet reception [26]. Different network topologies such as star, hierarchical/tree or mesh networks affect the reliability of the network and of the system respectively. Other factors which affect the reliability of a link are the transmission range of the signal strength, noise, fading effects, interference, modulation method, and frequency. Reliability of wireless links can be classified into two main categories the deterministic models and the probabilistic ones.

One of the most important issues for a system designer is to validate system reliability and identify the weakest components in order to replace, redesign and find alternative solutions. System reliability depends on component's arrangements. The two basic arrangements which we are focused on are components in series and in parallel. Other arrangements can include parallel-series, k-out-of-n or non-series-parallel systems [19]. More specifically, for components *in series*, the reliability quickly decreases as the number of components increases. In a serial system a single failure results in entire assembly or system failure. The addition of new components in series decreases the reliability of system. Components in series may have arrangements either following the sequence or parallel-split workflow patterns. This occurs because a failure of a single component will result the failure of the system. Reliability of systems in series can be defined as follows:

Definition 1. *Let $C = \{C_1, C_2, ...C_n\}$ be a number of components in series and $R = \{R_1, R_2, ..., Rn\}$ be the reliability of each component, then the component composition C will have reliability equal to:*

$$R = \prod_{k=1}^{n} (R_k) \tag{1}$$

In components *in parallel*, the reliability of the system exists only when at least one component is functional. The reliability of the system is the 1 minus the probability that all fail. In parallel components, all redundant units failure causes system failure. Thus, the addition of components in parallel increases the reliability of the subsystem. We may associate the multi-choice pattern as a parallel arrangement because the failure of a single component does not cause system failure. Reliability of components in parallel can be defined as follows:

Definition 2. *Let $C = \{C_1, C_2, ...C_n\}$ be a number of components in parallel and $R = \{R_1, R_2, ..., Rn\}$ be the reliability of each component, then the parallel component composition C will have reliability:*

$$R = 1 - \prod_{k=1}^{n} (1 - R_k) \tag{2}$$

Based on the above definitions, it can be easily proven that when C is the composition of components C_1 and C_2 with reliability R_1 and R_2 respectively, then (a) if a serial composition C preserves the reliability property, both C_1 and C_2 will satisfy the reliability property and (b) if both C_1 and C_2 preserve the reliability property, the parallel composition C will also satisfy the reliability property. Let assume that a system with two placeholders in series provides the reliability property. Then as defined above both C_1 and C_2 should provide the reliability property given $C = C_1 \cup C_2$. If there is no atomic component to preserve the above reliability of C_1, then a parallel composition of C_{1_1} and C_{1_2}: $C_1 = C_{1_1} \cap C_{1_2}$ could provide such property. The same procedure can be followed for C_2 as well. Based on such parallel composition, we are able to create a reliable composition of atomic components. The procedure of designing CPS with respect to reliability property is encoded as a pattern and it is presented in the next section.

IV. Reliable Component Composition Patterns

In order to design CPS with respect to S&D properties a pattern language should be defined. Design patterns can give solutions to problems by the use of formal proven properties. More specifically, the pattern should be able to define compositions of complex CPS to guarantee the required S&D property at design level. Based on these patterns a designer will be able to construct CPS architectures without the need to prove previously verified S&D properties. Reliable Component Composition Patterns (RCC) can be used for the discovery of component compositions with verified reliability properties. An RCC pattern specifies the order of component compositions constituting a primitive component orchestration (sequential or parallel compositions) and the data flows between them. It also specifies rules that dictate the properties that the constituent components must have. RCC

patterns are composed of: (a) an abstract workflow structure, defining the control structure and data flows of components, (b) the reliability property that the pattern guarantees and (c) the required reliability of compositions in order to guarantee *(b)*. Based on the previously proven reliability property, we may define:

Definition 3. *If a system H contains $H_1, H_2, ..., H_n$ placeholders and R_r is the required reliability property for H, R_r can be guaranteed if the reliability property R of the system satisfies the condition $R_r \leq R$.*

More precisely, an RCC pattern is able to validate system reliability and in case that the property is not guaranteed, it adds or replaces current components with other atomic ones or compositions in order to guarantee system reliability. In Figure 2 the execution order of pattern is depicted. First, the pattern validates whether the serial composition satisfies the required reliability property. If this property is not satisfied, a component is added in parallel in order to increase individual component reliability. The procedure continues until the composition of all components guarantees the required property.

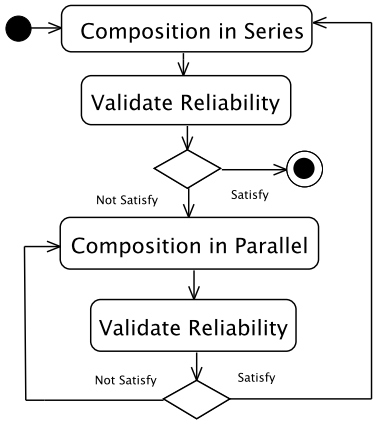

Fig. 2. Activity diagram of the RCC pattern

RCC patterns can be expressed as rules in Drools production system [4]. Drools rule engine supports backward and forward chaining inference by implementing and extending the Rete algorithm [27]. Drools production rules are stored in the production memory and are used to process data inserted in the working memory (Knowledge Base) as facts by pattern matching. Each rule consists of two parts: the "when" condition and the "then" actions. When the conditions of a rule in the Left Hand Side (LHS) are satisfied, then the rule is fired executing the actions as described in the Right Hand Side (RHS). In the RHS, facts can be inserted, updated or deleted in the knowledge base. RCC patterns encode compositions in Drools corresponding to the structure of the logical reliability arrangements. The main target of RCC patterns is to find suitable component compositions in order to guarantee the required reliability property. When system reliability does not satisfy the required reliability property, the pattern will have to substitute or add new atomic components able to guarantee the required reliability property. If a component with the required reliability does not exist in the knowledge base, a component composition shall be created.

TABLE I
RELIABILITY RULES FOR COMPONENT COMPOSITIONS

```
1 rule "Serial Reliable Component Composition Rule"
2   when
3     $C1:= Component($v1:=input, $v2:= output, $R1:= rel)
4     $C2:= Component($v2:=input, $v3:= output, $R2:= rel)
5     $P := Property(type=="Reliability", $R:= rel, $R<=$R1*$R2)
6   then
7     insert( new Component($v1,$v3, $R1*$R2,"reliable"));
8     if ($v1.type=="source" && $v3.type=="sink"){
9         retract($P);
10    }
11 end
12
13 rule "Parallel Reliable Component Composition Rule"
14   when
15     $C1:=Component($v1:=input, $v2:= output, $R1:= rel)
16     $C2:=Component($v1:=input, $v2:= output, $R2:= rel)
17     $P :=Property(type=="Reliability",$R:=rel,$R<=$R1+$R2-$R1*$R2)
18   then
19     insert( new Component($v1,$v2,$R1+$R2-$R1*$R2,"reliable"));
20 end
```

We may consider two components C_1 and C_2 having as source inputs I_1, I_2, sink outputs O_1, O_2 and reliability R_1, R_2. The composition of C_1 and C_2 will be described as a new component C with reliability R based on the components' arrangement. For the component composition in series, the control flow describes the serial arrangement of the components C_1 and C_2. The data flow defines that for the C_1 the output O_1 should be the input of C_2. The composition C will have as an input the $I = I_1$ and as an output the $O = O_2$. In addition, the reliability property guaranteed by a serial component composition is equal to $R = R_1 \cdot R_2$. Therefore, the guaranteed reliability property R should satisfy the required reliability property $R_r \leq R$. The encoded pattern in Drools is depicted in Table I. The *Serial Reliable Component Composition Rule* of RCC pattern defines three processes of the pattern: the composition of the components, the validation and the guarantee concerning the reliability of the serial component composition. In the LHS, the rule searches for suitable components in which the output of C_1 will be the input of C_2 in order to define a serial composition *lines 3-4*. In *line 5* the required reliability property is given. If the condition of $R \leq R1 \cdot R2$ is met, the rule enters in the RHS. In the RHS, the rule creates a new component C with the input of C_1 and the output of C_2, and as reliability the product of R_1 and R_2 *line 7*. The chosen stamp *reliable* defines a reliable component composition. The new component C is inserted in the working memory. In *line 8-10* the rule checks whether the input is the source and the output is the sink of the workflow. If the condition is met, the rule will retract the required property P indicating the end of the procedure because the pattern succeeds its goal. If not, the rule will be fired again based on the inserted component as a new fact. If the property is not satisfied then the components C_1 and C_2 should be replaced by a new component composition in parallel. The *Parallel Reliable Component Composition Rule* is then fired to find

new components which guarantee the required property. The control flow of this rule defines the multi-choice selection of components C_1 and C_2. The data flow of the component C_1 (*line 15*) which is in parallel with the C_2 (*line 16*) should have as input the $I_1 = I_2$ which is also the input for C_2 and as a output $O_1 = O_2$. The reliability property which should be guaranteed by a parallel component composition is equal to $R \leq R_1 + R_2 - R_1 \cdot R_2$ (*line 17*). If the condition is met, the parallel component composition C will be inserted in the working memory as a new component having as an input $I_0 = I_1 = I_2$, as an output $O_0 = I_1 = I_2$ and as reliability the $R_1 + R_2 - R_1 \cdot R_2$ (*line 19*).

RCC patterns for non-series-parallel and k-out-of-n logical arrangements can be expressed using this approach but they are not discussed due to space limitations. To extend our approach for complex systems including spanning tree component compositions, a depth-first search based on a graph theory approach can be used adequately. Especially for control flow analysis, a reverse postordering depth-first search can be used to produce natural linearisation of directed graphs.

V. Application of RCC Patterns in CPS: An Example

RCC patterns can be used for the design of reliable CPS. Component composition is applied in such a manner (a) to utilize specific functions and (b) to satisfy the reliability property of the system. Based on RCC patterns, we are able to design architectures for specific CPS applications. To evaluate the performance of RCC pattern, we define an application in which the pattern can be used satisfactorily. We may consider as an example, the design of a wireless sensor network attached to a physical architecture to send monitored data to a central controller through relay nodes and paths. The system reliability is related to the reliability of cyber and physical components and to the connectivity between these components. Failures or attacks on the wireless network consisting of wireless sensors may have as a result that possible anomalies in the CPS cannot be transmitted to the central controller.

A. Discussion

We consider as a source the location of a monitoring mechanism and as a sink the location of a central controller. The problem to be resolved regards the identification of the number and the location of relay sensors that should be installed in order to ensure a reliable CPS monitoring network. The solution includes the design of a reliable monitoring system, based on RCC patterns.The inputs of this application include the distance between source and sink, the transmission range of sensors, the reliability of the components and the required system reliability. The outputs include the number of sensors, their location and the number of paths and hops, in order to satisfy the required reliability property. The accuracy and the reliability of the monitoring mechanism depend on the reliability of the components which compose the system.

Let assume that the distance between the source and the sink is L. The distance between two wireless nodes is related to the transmission range of the two nodes and it can be found from the Friis transmission equation [28]. If the maximum distance between two wireless sensors is d, the minimum number of relay nodes can be calculated from $P = L/d - 1$. Based on this number, we may assign the placeholders of the system in where nodes should be installed in series. We may consider a deterministic approach for the connectivity between two sensors. If the distance between the two nodes is greater than the maximum distance d then the reliability of the wireless link is 0. On the other hand, if the distance is less or equal than d, the link reliability is 1. This is an assumption to our approach because in reality the reliability of a wireless link is probabilistic in where other factors such as interference, path loss and propagation can influence the reliability of the connectivity. This is an interesting topic for research but it is out of the scope of this paper.

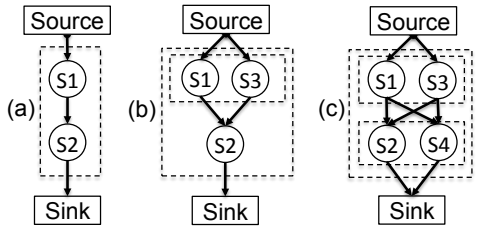

Fig. 3. Design phases of a sensor network with reliability (a) 96% (b) 98% (c) 99.9%

Back to our example, we may consider a system with P placeholders where in each placeholder a sensor should be installed. The reliability of the system will be equal to the composition of sensors in series: $R = \prod_{k=1}^{n}(R_k)$. The RCC pattern is able to validate whether the reliability of the system satisfies the required reliability and if not it will add sensors in parallel in order to guarantee the required reliability. This will produce solutions concerning the number of sensors and their location. Let assume that a system includes 2 placeholders P_1 and P_2, the reliability of each sensor is 98% and the required reliability is 99%. The pattern will validate system reliability by placing two sensors S_1 and S_2 in series at placeholders. Calculating the reliability of the system it will give: $R = R_1 \cdot R_2 = 0.98 \cdot 0.98 = 0.96$ which is lower than the required reliability. Since the reliability is not guaranteed, the pattern will add an S_3 at placeholder P_1. A new validation occurs giving: $R = (1-(1-R_1)\cdot(1-R_3)\cdot R_2 = 0.98$. The reliability is close to 99% but even now the property is not satisfied. Therefore, the pattern will add a new sensor in placeholder P_2. Finally, the $R = (1-(1-R_1)\cdot(1-R_3)\cdot(1-(1-R_2)\cdot(1-R_4) = 0.999$. The described procedure is depicted in Figure 3. This example shows the procedure which is followed for designing a reliable system containing 2 placeholders. However, in case of multi-hops networks the solution is not so easily provided. RCC pattern is able to provide solutions for multi-hops networks as presented in the next subsection.

B. Implementation and Experimental Results

To give a proof of concept of our approach, we implement the described scenarios in the Eclipse Modeling Tool (4.4.1) with the JBoss Drools 6.2.0.CR3 extension. Each component is defined as a Java class corresponding to the components of the system. The previously described RCC pattern, as expressed in Drools rules, is used for constructing a reliable monitoring network consisting of wireless sensors. The number of placeholders is based on the distance between the source and the sink node and the wireless transmission range of each sensor. Let assume that each sensor has reliability factor 98% and transmission range $100m$. As different factors of the experiments, we may consider the distance between the source and sink, which reflects the number of placeholders based on $P = L/d - 1$. The experiments were conducted on an Intel Core i7 with 8Gb RAM. The number of sensors, the execution times for different distances between source and sink of the described scenario are presented in Table II. As we can observe from the results, the execution time is increased as the number of distance is increased. In addition the number of sensors necessary for preserving the required reliability property is growing exponentially. The described pattern-based methodology can be extended to design more complex CPS architectures, such as cyclic compositions, by the addition of proven patterns as rules for different S&D properties.

TABLE II
RESULTS OF CONDUCTED EXPERIMENTS

Distance (metres)	Number of Placeholders	Reliability	Number of Sensors	Execution Time (msec)
1.000	9	99.6%	18	17
2.000	19	99.2%	38	19
5.000	49	99.0%	148	47
7.000	69	99.9%	376	58
10.000	99	99.0%	548	78

VI. CONCLUSIONS AND FUTURE STEPS

In this work a pattern-based approach for designing reliable CPS is presented. The main idea of our approach is the design of CPS with respect to S&D properties. Our work includes a methodology on how to preserve a S&D property through a pattern, encoded as rule-based reasoning, for designing reliable architectures of complex CPS. Reliable Component Composition patterns have been defined to describe the order of the execution and the data flow between placeholders, to validate reliability of the compositions and to guarantee system reliability. To evaluate our proposed scheme, RCC patterns were used in order to design a reliable monitoring mechanism consisting of a wireless sensor network. Our future steps include the extension of our approach to cover other S&D properties which are also critical for the design of complex CPS. This pattern-based language will be used for the development of a framework, covering not only horizontally layered designs but also vertical ones for the design of CPS systems preserving required S&D properties.

REFERENCES

[1] J. Sztipanovits and et al. Strategic R&D Opportunities for 21st Century Cyber-Physical Systems. *Steeting Commitee for the Foundation fo Innovation in Cyber-Physical Systems, National Institute of Standards and Technology (NIST)*, 2013.

[2] E. Lee. Cyber physical systems: Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369. IEEE, 2008.

[3] N. Petroulakis, I. Askoxylakis, A. Traganitis, and G. Spanoudakis. A privacy-level model of user-centric cyber-physical systems. In *Human Aspects of Information Security, Privacy, and Trust*, 2013.

[4] Drools. Business rules gement system solution. www.drools.org.

[5] J. Lin, S. Sedigh, and A. Miller. Modeling cyber-physical systems with semantic agents. In *Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual*. IEEE, 2010.

[6] P. Derler, E. Lee, and A. Sangiovanni-Vincentelli. Modeling cyber–physical systems. *Proceedings of the IEEE*, 100(1):13–28, 2012.

[7] DC. Schmidt. Model-driven engineering. *Computer Society*, pages 286–298, 2006.

[8] T. Kurpick, C. Pinkernell, M. Look, and B. Rumpe. Modeling cyber-physical systems: model-driven specification of energy efficient buildings. In *Modelling of the Physical World Workshop*. ACM, 2012.

[9] H. Petritsch. Service-oriented architecture (soa) vs. component based architecture. *Vienna University of Technology white paper, available at http://whitepapers. techrepublic. com. com/abstract. aspx*, 2006.

[10] G. Gössler and J. Sifakis. Composition for component-based modeling. *Science of Computer Programming*, 2005.

[11] G. Simko, D. Lindecker, T. Levendovszky, S. Neema, and J. Sztipanovits. Specification of Cyber-Physical Components with Formal Semantics Integration and Composition. pages 471–487, 2013.

[12] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994.

[13] G. Spanoudakis and S. Kokolakis. *Security and Dependability for Ambient Intelligence*. Springer Science & Business Media, 2009.

[14] A. Maña, E. Damiani, S. Gürguens, and G. Spanoudakis. Extensions to pattern formats for cyber physical systems. In *Proceedings of the 31st Conference on Pattern Languages of Programs (PLoP14)*, 2014.

[15] L. Pino, K. Mahbub, and G. Spanoudakis. Designing Secure Service Workflows in BPEL. In *12th International Conference, ICSOC 2014, Paris, France, November 3-6, 2014.*, 2014.

[16] L. Pino, G. Spanoudakis, A. Fuchs, and S. Gürgens. Discovering secure service compositions. In *4th International Conference on Cloud Computing and Services Sciences, Barcelona, Spain*, 2014.

[17] B. Douglass. *Real-time design patterns: robust scalable architecture for real-time systems*, volume 1. Addison-Wesley Professional, 2003.

[18] C. Singh and A. Sprintson. Reliability assurance of cyber-physical power systems. In *Power and Energy Society General Meeting*. IEEE, 2010.

[19] D. Coit and A. Smith. Reliability optimization of series-parallel systems using a genetic algorithm. *IEEE Transactions on Reliability*, 45(2), 1996.

[20] E. Elsayed. *Reliability engineering*. Wiley Publishing, 2012.

[21] L. Sha and J. Meseguer. Design of complex cyber physical systems with formalized architectural patterns. In *Software-Intensive Systems and New Computing Paradigms*, pages 92–100. Springer, 2008.

[22] L. Coppolino, L. Romano, N. Mazzocca, and S. Salvi. Web services workflow reliability estimation through reliability patterns. In *SecureComm*. IEEE, 2007.

[23] W. Van Der Aalst, A. Ter Hofstede, B. Kiepuszewski, and A. Barros. Workflow patterns. *Distributed and parallel databases*, 14(1), 2003.

[24] A. Geraci, F. Katki, L. McMonegal, B. Meyer, J. Lane, P. Wilson, J. Radatz, M. Yee, H. Porteous, and F. Springsteel. *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. 1991.

[25] J. Laprie. Dependable computing and fault-tolerance. *Digest of Papers FTCS-15*, pages 2–11, 1985.

[26] P. Park, P. Di Marco, C. Fischione, and K. Johansson. Modeling and optimization of the ieee 802.15. 4 protocol for reliable and timely communications. *Parallel and Distributed Systems*, 24(3), 2013.

[27] C. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial intelligence*, 19(1):17–37, 1982.

[28] H. Friis. A note on a simple transmission formula. *IRE*, 34(5), 1946.