Proceedings of the IASTED International Conference
Modelling, Identification and Control (MIC 2014)
February 17 - 19, 2014 Innsbruck, Austria

DESIGNING RESILIENCE MEDIATORS FOR CONTROL SYSTEMS

Mauno Rönkkö
Department of Environmental Science
University of Eastern Finland
PO Box 1627
70211 Kuopio, Finland
mauno.ronkko@uef.fi

Markus Stocker
Department of Environmental Science
University of Eastern Finland
PO Box 1627
70211 Kuopio, Finland
markus.stocker@uef.fi

Mats Neovius
Department of Information Technologies
Åbo Akademi University
Joukahaisenkatu 3-5A
20520 Turku, Finland
mneovius@abo.fi

Luigia Petre
Department of Information Technologies
Åbo Akademi University
Joukahaisenkatu 3-5A
20520 Turku, Finland
luigia.petre@abo.fi

Mikko Kolehmainen
Department of Environmental Science
University of Eastern Finland
PO Box 1627
70211 Kuopio, Finland
mikko.kolehmainen@uef.fi

ABSTRACT
In this article we discuss how to improve the resilience of an existing control system. In recent years, our environment has become populated with numerous control systems due the to availability of low-cost technologies. For instance, modern home automation has become a cooperative network of multiple control systems, many of which communicate over the Internet. Many of these systems hardly address resilience, and improving them is hard, as many of them are provided as "black boxes". Consequently, as the main contribution, we propose a method for introducing resilience to an existing control system. The method is based on designing and adding resilience mediators that act in between the components to correct faulty communication and to mediate state awareness. In the method, behavioral analysis and HAZOP tables are used as tools to identify and design the relevant resilience mediators. We illustrate the use of the proposed method with an ongoing, real-life case study involving the control of a residential building that adapts to occupant's behavior.

KEY WORDS
Intelligent Control, Robust Control, Resilient Control, Mediators, Sequence diagrams, HAZOP

1 Introduction

Traditionally, control systems are seen as closed systems with well defined boundaries. For closed systems, one can design alternative control strategies and choose the best for the purpose. Depending on how critical the control system is, one can also opt for control strategies that provide resilience against faults and undesired dynamics. Such systems are generally known as resilient control systems [1].

Recently, however, a new trend has emerged where non-critical control systems are connected to other components via the Internet. These components can be sensors, data services, actuators, or even other control systems. For instance, modern home automation builds upon wireless and mobile sensors and devices that depend on cloud services. Consequently, the aggregated systems include evermore frequently information services and data sources that are not built with control in mind. Such external services are prone to faults and faulty behavior which are often unknown till it emerges. For instance, in case of home automation, a weather service provides forecasts that are used to optimize heating and cooling and thereby save energy and costs. However, a weather service may become temporarily unavailable or, even worse, it could suddenly provide faulty forecasts due to some communication error.

Improving the resilience of aggregated systems is by no means trivial, as the external systems and services are often isolated "black boxes" and cannot be enhanced or refined in any way. Furthermore, the service interfaces of the external services, including the service logic, may change without a notice, leading to faulty messaging and unpredictable system's dynamics. Also, as the number of connected components grows, it becomes harder to track the cause for a fault. A fault may then propagate over communication from one component to another, causing the con-

DOI: 10.2316/P.2014.809-012

nected components to enter faulty states one by one. This cascading effect has been studied, for instance, by Zhu et al. [2]. It is also hard to resolve such an aggregated faulty state without shutting down some or all of the connected components. In some cases, shutting down a component may not even be an option.

With this problem setting in mind, and as the main contribution, we propose here a method for designing and implementing resilience mediators. A resilience mediator is a data mediator [3] that sits in between two components, relaying their messages back and forth. The resilience mediator maintains state awareness [1] and corrects or refines the messages accordingly before relaying them from one component to another. In this way, propagation of faults becomes restricted while requiring no changes to the original components. Thus, we focus here on resilience issues that can be identified from the communication and that can also be mediated by enhancing the communication between the connected components. In this sense, the proposed method is complementary standard design methods and to existing work on resilient control systems, such as [4, 5, 6], where differential equations are used to solve the control of the actual physical processes, and statistical methods and fuzzy logic are used to improve the performance.

To design a resilience mediator, we propose here a method where behavioral analysis is used first to uncover critical messaging. More specifically, we use sequence diagrams for this purpose. Then, HAZOP tables [7] are used to analyze how the identified messaging can deviate. With HAZOP tables, we can identify not only causes and consequences for potential faults, but also possible indicators, safeguards and corrective actions. After HAZOP analysis, we can decide what resilience mediators are needed and which faults and faulty states they should mediate. To help constructing the mediation logic, we have implemented a simple tool for mapping an identified fault to a mediated fault state. We have also implemented another tool to simulate and test the mediation logic under multiple simultaneous faults and deviations.

We illustrate the method with a real-life case study involving home automation. The case study is part of our ongoing research on adaptive home automation. We analyze how a home automation system could be made more resilient in case of adaptive dynamics, where it adapts heating and ventilation according to weather forecasts to save costs, especially when an occupant is on vacation. The case study shows how even in such a simple setting there are many critical faults that need to be considered. It also shows how specifically designed resilience mediators can significantly improve the overall stability of the system even under multiple critical faults and deviations.

The article in organized as follows. In Section 2, we discuss control systems and resilience mediators in more detail. In Section 3, we present the proposed method to design resilience mediators. In Section 4, we present the case study of an adaptive house. In Section 5, we present the results. Finally, in Section 6, follows the conclusion.

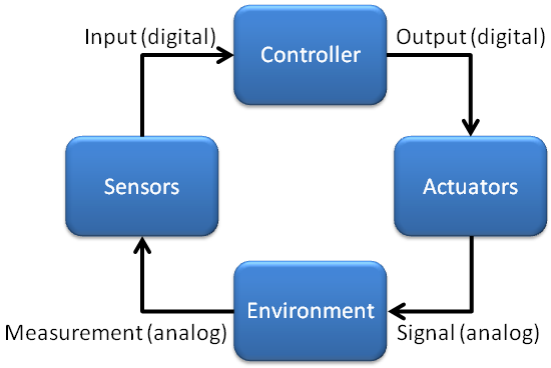

Figure 1. Traditional control system with a digital controller.

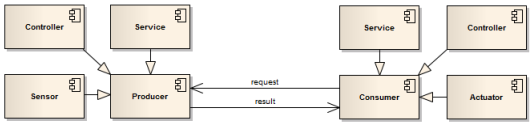

Figure 2. Systems view of a control system.

2 Control Systems and Resilience Mediators

Figure 1 depicts a traditional control system architecture [8]. In the architecture, the controller is digital and both the sensors and the actuators perform conversions between digital and analog signals. The control logic is implemented within the controller. In addition, the sensors and the actuators may consist of local intelligence supporting the implementation of the control logic on a higher level of abstraction.

As discussed in [8], however, the traditional control system architecture is limited when considering modern control systems that are connected over the Internet. As the connected components may be services and data sources that are not built with control in mind, we consider here a more generic systems view, as depicted in Figure 2. In the systems view, a control system is seen as a specific case of a producer-consumer pattern, where the producer not only captures sensors but also services and even the output of other controllers. Likewise, the consumer captures not only actuators and the control logic but also some service logic.

Typically, any resilience improvements are implemented directly to the relevant components. This, however, is not sustainable, as the resilience logic becomes scattered among components. The reason for this is that resilience improvements hardly ever localize to a single component. Even the detection of a resilience issue requires communication between multiple sensing and monitoring components. Because of this, control system updates become gradually harder.

To simplify and to unify the implementation of re-

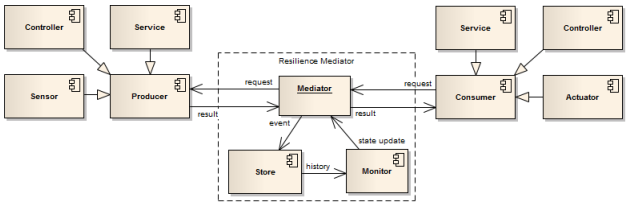

Figure 3. Systems view of a control system with a resilience mediator.

silience improvements, we consider control systems with resilience mediators, as depicted in Figure 3. A resilience mediator is a data mediator [3]. It is placed in between the producer and consumer components relaying their messages. Based on the messages, the mediator detects and stores events. Stored events are inspected by a monitor that notifies the mediator about state changes. In this way, the resilience mediator can maintain state awareness [1] and correct or refine a message between the producer and consumer components accordingly. Moreover, this approach ensures that all resilience improvements localize to the resilience mediators that coordinate the resilience actions, making the implementation of later improvements more manageable.

In this article, we focus on the design of the mediation logic that includes state awareness for the resilience mediators. Thus, we do not consider the actual correction or modification of a faulty message passed through the mediator. Such activity is always case specific and it is nevertheless governed by the mediation logic.

3 Method to Design a Resilience Mediator

Next, we present the method to design a resilience mediator. As depicted in Figure 4, there are five steps in the design workflow. We shall now briefly describe these steps. The steps are illustrated later with a case study. Note that there is a natural feedback loop between the steps, as indicated in Figure 4. However, for clarity, we present next the steps without feedback.

Behavioral analysis. The first step is a behavioral analysis of the system with controllers, sensors, and actuators. The objective is to capture the essential communication between the components with respect to specific events of interest. For this purpose, we use standard sequence diagrams [9].

Deviation analysis. In the second step, each sequence diagram is analyzed for deviations. More specifically, as we focus on the communication activity, each arrow in each sequence diagram is analyzed for possible deviations with causes and consequences. The analysis results are then captured using HAZOP tables [7]. Thus, for each arrow in the

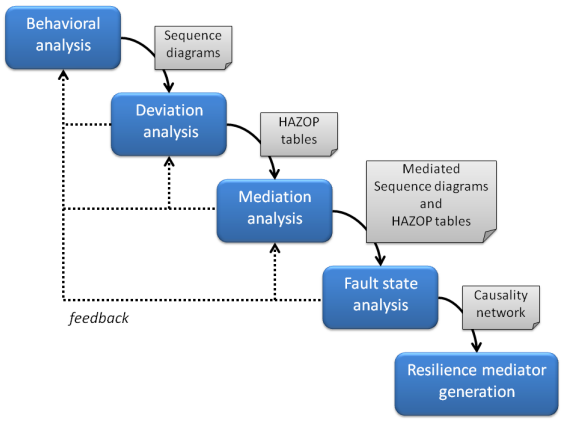

Figure 4. Workflow of the method to design resilience mediators.

sequence diagram there is a HAZOP table capturing deviations of interest, their causes and consequences, potential indicators, safeguards, and corrective actions.

Mediation analysis. In the third step, the HAZOP tables are studied to identify deviations that can be mediated with a resilience mediator. A deviation can be mediated if it can be identified from the communication messages and the faulty communication can either be corrected or augmented to avoid fault propagation. Based on the analysis, the sequence diagram is updated by adding mediators and their communication for the identified deviations. In addition, for each added communication arrow, a new HAZOP table is added as in the previous step. Note that the purpose of the resilience mediator is to restrict fault propagation by correction, refining, or augmenting a faulty or deviating message. If the added HAZOP tables do not confirm this for some deviations, they cannot be mediated.

Fault state analysis. In the fourth step, fault propagation is analyzed. First, as a single cause can result in multiple deviations, fault propagation within the components is analyzed from the HAZOP tables by studying the relation between causes and consequences. The result is a set of rules and dependencies that describes what deviations in HAZOP tables arise as a consequence of other deviations within the same component. Next, the set of rules is extended by adding new rules that link faulty communication of each component to some state in a mediator. Consequently, any faulty communication that cannot be linked to a mediator state requires human intervention. Similarly, a mediator state that is not linked to any faulty communication is superfluous. All communication sent by a mediator must be free of faults. In other words, a mediator is allowed to send non-optimal messages, but a sent message cannot cause any deviations in the receiving component. All these rules form a causality network describing how HAZOP de-

viations are linked to each other within components and how faulty communication is mediated by the mediators. Each rule is represented as a logical implication. Formally, a deviation regarding an activity of a component is seen here as a triple D:

$$D \hat{=} (component, activity, deviation)$$

Thus, a set of n deviations is captured as S:

$$S \hat{=} \{(c_1, a_1, d_1), ..., (c_n, a_n, d_n)\}$$

Consequently, we denote the set of deviations S caused by an initial deviation D as the mapping M:

$$M \hat{=} D \rightarrow S$$

Thus, the HAZOP tables determine a set of m mappings that form in this way the causality network N:

$$N \hat{=} \{M_1, ..., M_m\}$$

Technically, the causality network N forms a directed asyclic graph.

Resilience mediator generation. In the fifth and the last step, the derived causality network is used to generate the core logic of the resilience mediators. This can be done because the causality network is captured as a set of logical implications. Obviously, the generated core logic needs then to be augmented with algorithms and methods that actually do the identification of faults and deviations as well as the correction or refinement of faulty communication. However, the generated core logic can be used to test and verify the functioning of the overall communication logic before starting the implementation of the missing algorithms and methods.

Some remarks on deviation detection and correction algorithms. The proposed method for designing resilience mediators considers only fault propagation and state awareness. Therefore, it focuses only on the design of mediation logic and mapping of faults to appropriate recovery mechanisms within the mediators. Because of this, expert knowledge is required to implement the deviation detection and correction algorithms for the designed mediators. The implementation of the algorithms may, thus, require use of artificial neural networks and data mining methods. The algorithms may also use semantic methods, including ontologies, to realize the mapping between knowledge, concepts, and data values. Also, the algorithms do not need to operate in isolation; they may use external services, for instance, to perform computationally demanding analysis. In extreme cases, the algorithms can also contact maintenance and require human interaction. For instance, a deviation correction algorithm could send an SMS message to the occupant requesting a repair.

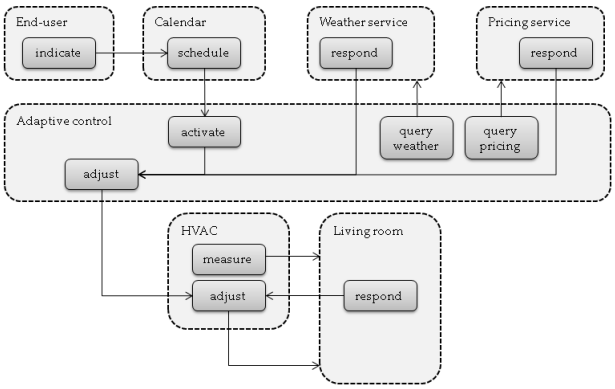

Figure 5. Components of an adaptive house relevant to the case study.

4 Case Study: An Adaptive House

Recently there has been a growing interest on researching optimality and safety of home automation under varying external conditions. For instance, Aswani et al. [10] have studied reduction of electricity consumption using learning-based methods. Raatikainen et al. [11] have studied energy efficiency by using SOM, and Skön et al. [12] have studied indoor air quality by using SOM. There are also studies on using semantic technologies [13] to support the interpretation of residential building measurements. All this can be seen as research leading eventually to the development of an adaptive house. It adjusts home automation and HVAC based on learned and predictive dynamics. It uses various pricing and forecasting services to optimize energy efficiency while maintaining healthy living.

In this article, we consider as the case study an adaptive house with components and activities as depicted in Figure 5. For clarity, we shall focus here only on one event: the adaptation of home automation to occupants' vacation. During their vacation, the occupants are assumed to be away from home. Technically, the event starts when the occupants enter their vacation to a calendar. The adaptive house uses the entry for instance to plan heating, cooling, and ventilation of an empty house. Its objective is to save energy during the vacation while maintaining nominal living conditions. The living conditions are restored at the end of the vacation, so that when the occupants arrive, the house has optimal living conditions. In order to do this, the adaptive house accesses not only the calendar, but also a weather forecast service as well as relevant pricing services regarding electricity, heating, cooling, and ventilation.

The case study is part of ongoing research with our living lab, called AsTEKa [14]. AsTEKa is a sensor network for monitoring indoor air quality and energy efficiency of residential buildings. It measures and monitors, for instance, CO_2 gas levels, room temperature, water consumption and use of electricity. The AsTEKa sensor network was incorporated and successfully tested in several

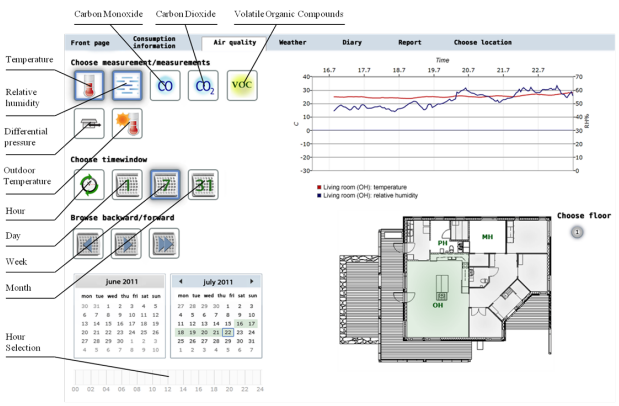

Figure 6. AsTEKa's user interface [15].

houses in the 2010 Kuopio Housing Fair. Currently, it continuously measures and monitors more than 30 locations, including offices and school buildings. A snapshot of AsTEKa's user interface is depicted in Figure 6.

5 Results

We shall now discuss results obtained with the proposed method on the adaptive house case study. The discussion proceeds in the same order as the steps in the proposed method.

Behavioral analysis. In this first step, we performed a behavioral analysis on Figure 5 with respect to the vacation event as presented earlier. This resulted in a sequence diagram depicted in Figure 7. The diagram captures the identified, essential communication between the components with respect to the vacation event. There are also two loops in the diagram capturing the adaptive control loop and the standard HVAC control loop during the event. In the diagram, the occupant initiates the event by adding an entry of the vacation to the calendar.

Deviation analysis. In the second step, we constructed a HAZOP table for each communication activity of the sequence diagram of Figure 7. In the diagram, a communication activity is represented with an arrow. All HAZOP tables were constructed by using the same structure which is shown in Figure 8. Thus, we constructed a total of 11 HAZOP tables capturing the 11 arrows of Figure 7. Figure 9 shows an example of a HAZOP table capturing the second arrow of the sequence diagram, "schedule(vacation)". It captures the vacation scheduling event initiated by the calendar for the adaptive control. For this communication activity, we identified six deviations of interest as shown in Figure 9. For three of them there are no safeguards, but they all share the same deviation indicators: motion detectors and use of appliances.

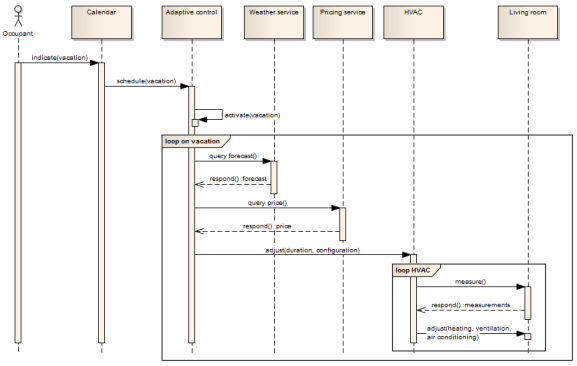

Figure 7. Sequence diagram of the case study.

Component				HAZOP	
Activity				Author	
Intention				Date	
Deviation	**Cause**	**Consequence**	**Indicator**	**Safeguard**	**Correction**

Figure 8. The generic structure of a HAZOP table.

Component	calendar			HAZOP	OnVacation
Activity	schedule			Author	Mauno Rönkkö
Intention	schedules vacation period			Date	August 28, 2013
Deviation	**Cause**	**Consequence**	**Indicator**	**Safeguard**	**Correction**
missing	entry missing	not entering vacation mode	motion detectors, use of appliances	none	add calendar entry
	calendar is faulty	"	"	"	repair
faulty	"	"	"	"	
too late	incorrect entry	"	"	"	extend calendar
	calendar is faulty	"	"	"	repair
too short	incorrect entry	exits vacation mode too early	"	confirm duration when entering vacation mode	extend calendar entry
	calendar is faulty	"	"	"	repair
too early	incorrect entry	enters vacation mode too early	"	confirm mode switch	postpone calendar entry
	calendar is faulty	"	"	"	repair
too long	incorrect entry	stays in vacation mode for too long	"	confirm duration when entering vacation mode	shorten calendar entry
	calendar is faulty	"	"	"	repair

Figure 9. Example HAZOP table of the vacation scheduling event initiated by the calendar.

Mediation analysis. In the third step, the mediation analysis revealed that the original sequence diagram in Figure 7 can be improved by introducing four resilience mediators: scheduling mediator, query mediator, adjustment mediator, and measurement mediator. The updated sequence diagram with added communication activities is depicted in Figure 10. Based on the sequence diagram, we also updated the component diagram shown earlier in Figure 5 by adding the four new resilience mediators. The updated component diagram is shown in Figure 11.

After updating the diagrams, we constructed new HAZOP tables for the new communication activities introduced to the sequence diagram along with the resilience mediators. There are all in all 9 new arrows in the updated sequence diagram shown in Figure 10. However, only 6 of the arrows were unique. Consequently, we ended up having only 6 new and disjoint communication activities to be captured with HAZOP tables. These communication activities were scheduling activity of the scheduling mediator, query and response activity of the query mediator, measurement and response activity of the measurement mediator, and adjustment activity of the adjustment mediator. Thus, we constructed 6 new HAZOP tables respectively. As an example of a newly constructed HAZOP table, Figure 12 depicts analysis of the scheduling activity of the scheduling mediator. As the figure shows, the scheduling mediator distinguishes between two deviations, one for faulty scheduling and the other for missing or incorrect timing. Thus, for the mediator to work properly, it has to be able to detect these deviations from the scheduling messages received from the calendar. In addition, it has to be able to refine or correct the faulty scheduling messages, before passing them on to the adaptive control component.

Fault state analysis. In this fourth step, we analyzed the fault propagation inside each component as well as between all components by analyzing the connectivity of the deviations captured by the HAZOP tables. To do this, we developed a binder tool depicted in Figure 13. The tool reads all the HAZOP tables and constructs all possible deviations in the form of triples D, as defined earlier. The tool then assists in determining the mappings, formalized earlier as implications M. The mappings then determine the entire causality network which was formalized as set N. As shown in the user interface, in Figure 13, the tool allows the user to pick a deviation (the leftmost panel), choose components that are affected by the deviation (the topmost middle panel) and the affected activities (top bottommost middle panel). The tool shows on the rightmost panel all such possible deviations, from which the user can choose the affected deviations. For instance, in Figure 13, it is captured that the deviation where the scheduling activity of the calendar is too late causes the correction of the scheduling activity for the scheduling mediator. Thus, for each deviation D the tool always shows the complete mapping M. The tool also stores the entire causality network N for further use.

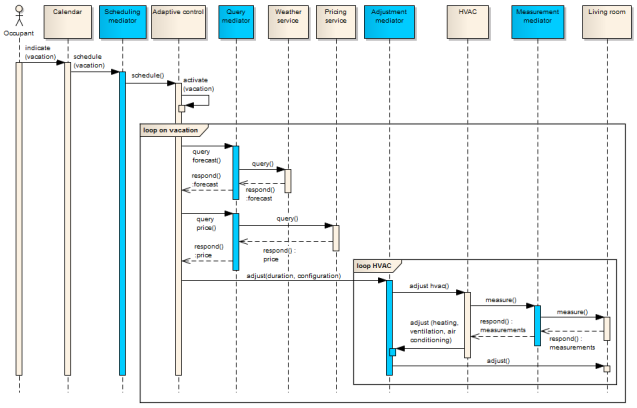

Figure 10. Updated sequence diagram with resilience mediators.

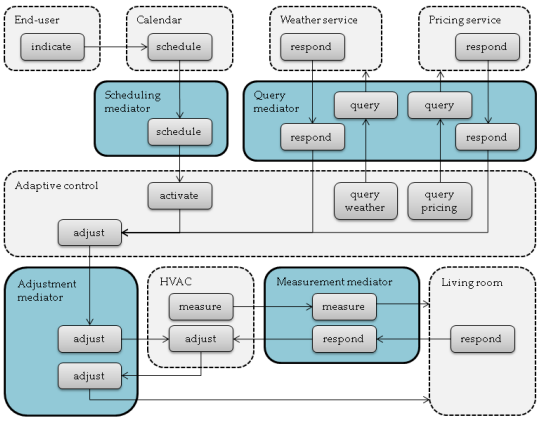

Figure 11. The adaptive house components with resilience mediators included.

Component	scheduling mediator	**HAZOP**	OnVacation
Activity	schedule	**Author**	Mauno Rönkkö
Intention	mediates scheduling	**Date**	August 28, 2013

Deviation	Cause	Consequence	Indicator	Safeguard	Correction
defaults	passed scheduling is faulty	non-optimal hvac	content analysis	none	repair
corrected timing	entry is missing or has incorrect timing	"	motion detectors and use of	confirm when entering or exiting onVacation mode	update calendar

Figure 12. Example HAZOP table of the scheduling activity of the scheduling mediator.

Because the mapping of a deviation to consequent deviations is defined to be transitive, we only need to define the mapping for the immediate consequent deviations. Any transitive consequences will follow by implication. This greatly simplifies the mapping process. With this in mind, we have implemented several algorithms to the developed binder tool, to help analyzing the causality network. For instance, there is an algorithm in the tool that detects causality loops. This is to ensure that the causality network forms indeed a directed asyclic graph. Another algorithm lists all initial causes, that is, deviations that cause other deviations, but are never themselves caused by any other deviations. Similarly, there is an algorithm that lists all final consequences, that is, deviations that are caused by other deviations, but that never cause themselves any other deviations. The tool can also list the entire causality network, as depicted in Figure 14.

Resilience mediator generation. The fifth step is mechanical with respect to the mediation logic. In short, the causality network produced in the previous step can directly be used to generate the mediation logic for the resilience mediators. For instance, in our case study, the mediation logic regarding the scheduling mediator is shown in Figure 15. In the figure, the logic maps the scheduling deviations of the calendar to two fault states of the mediator: default and corrected timing. The default state forces the scheduling mediator to send default scheduling, whereas the corrected timing state causes the scheduling mediator to send a scheduling with corrected timing. For all this to work, we have to implement six algorithms for detecting calendar deviations: faulty scheduling, missing scheduling, too early scheduling, too late scheduling, too long scheduling, and too short scheduling. Correspondingly, we have to also implement two algorithms for correcting the deviation within the mediator: default scheduling for faulty schedule, and corrected timing for scheduling with incorrect timing. As noted earlier in the HAZOP table, in Figure 9, such algorithms can use motion detectors and state of appliances to deduce faults and proper corrective actions.

To help with the testing and validation of the generated mediation logic, we have implemented a tool that simulates the communication activity of the components based on the causality network. The tool is depicted in Figure 16. In short, the tool reads the causality network and let's the user inject and recover deviations. It then shows step by step, how deviations cause other deviations and how mediators react to them. For instance, Figure 16 captures the state after injecting the deviation $(calendar, schedule, tooearly)$. It causes the mediator to fall into the "corrected timing" state as seen in the rightmost panel. All the events are broadcast throughout the entire system, as seen on the bottom leftmost panel. The top leftmost panel is for injecting and recovering any possible deviation described in the causality network.

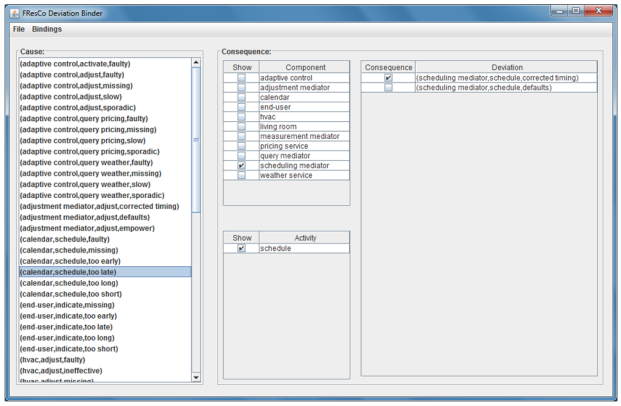

Figure 13. User interface of the binder tool for indicating causalities between deviations.

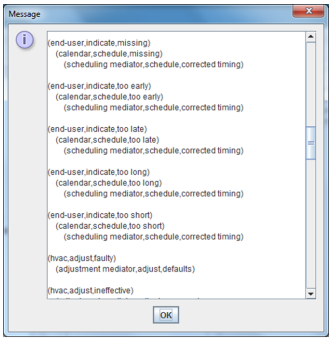

Figure 14. Listing of the causality network as shown by the binder tool.

(calendar,schedule,faulty) → (scheduling mediator,schedule,defaults)
(calendar,schedule,missing) → (scheduling mediator,schedule,corrected timing)
(calendar,schedule,too early) → (scheduling mediator,schedule,corrected timing)
(calendar,schedule,too late) → (scheduling mediator,schedule,corrected timing)
(calendar,schedule,too long) → (scheduling mediator,schedule,corrected timing)
(calendar,schedule,too short) → (scheduling mediator,schedule,corrected timing)

Figure 15. Medation logic of the scheduling mediator.

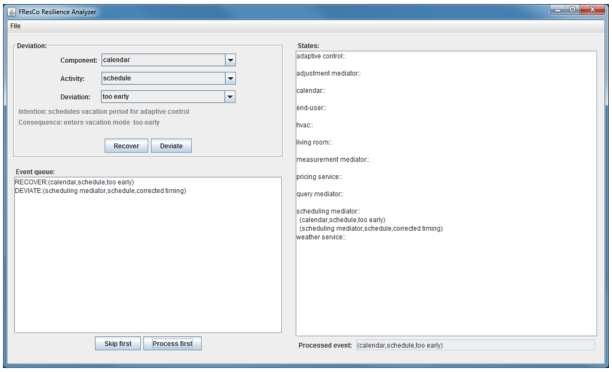

Figure 16. User interface of the analyzer tool for studying deviation propagation in the causality network.

6 Conclusion

In this article, we proposed a new method for designing resilience mediators for control systems. The method is composed of five steps: behavioral analysis, deviation analysis, mediation analysis, fault state analysis, and resilience mediator generation. The method uses sequence diagrams and HAZOP tables for the behavioral and deviation analysis, respectively. We have implemented a binder tool for fault state analysis and an analyzer tool for simulation and validation of generated mediation logic. We illustrated the method with a case of adaptive home automation, which is part of our ongoing research with an existing living lab.

As for future work, we plan to use an ontology based software framework, Wavellite [16], for simplifying the implementation of deviation detection algorithms. Wavellite supports the integration of computation methods and semantic reasoning, whereby it is an ideal framework for bridging the gap between HAZOP tables and the actual implementation of deviation detection algorithms.

Also, as future work, we plan to use a calculus for trustworthiness [17] to assess the trustworthiness of sensors, external services, and data delivered by them. This helps, for instance, in deciding if a sensor is showing signs of wear down, and if its value should be corrected by using supportive values from other sensors and data sources.

Acknowledgement

This research is funded by the Academy of Finland project "FResCo: High-quality Measurement Infrastructure for Future Resilient Control Systems" (Grant number 264060).

References

[1] C. G. Rieger, D. I. Gertman and M. A. McQueen, Resilient Control Systems: Next Generation Design Research, *Proc. of Human System Interactions*, 2009, 632-636.

[2] Q. Zhu and T. Basar, A dynamic game-theoretic approach to resilient control system design for cascading failures, *Proc. of International Conference on High Confidence Networked Systems* , 2012, 41-46.

[3] G. Wiederhold, Mediators in the architecture of future information systems, *Computer*, 25(3), 1992, 38-49.

[4] K. Ji and D. Wei, Resilient control for wireless networked control systems, *International Journal of Control, Automation and Systems*, 9(2), 2011, 285-293.

[5] O. Linda, M. Manic, J. Alves-Foss and T. Vollmer, Towards resilient critical infrastructures: Application of Type-2 Fuzzy Logic in embedded network security cyber sensor, *Proc. of Resilient Control Systems*, 2011, 26-32.

[6] Q. Zhu and T. Basar, Robust and resilient control design for cyber-physical systems with an application to power systems, *Proc. of Decision and Control and European Control Conference*, 2011, 4066-4071.

[7] F. Redmill, M. Chudleigh and J. Catmur, *System Safety:HAZOP and Software HAZOP*, (Wiley, 1999).

[8] W. Zhang, M. S. Branicky and S. M. Phillips, Stability of networked control systems, *Control Systems*, 21(1), 2001, 84-99.

[9] X. Li, Z. Liu and H. Jifeng, A formal semantics of UML sequence diagram, *In Proc. of Software Engineering Conference*, 2004, 168-177.

[10] A. Aswani, N. Master, J. Taneja, D. Culler and C. Tomlin, Reducing transient and steady state electricity consumption in hvac using learning-based model-predictive control, *Proc. of the IEEE*, 99(12), 2011, 1-14.

[11] M. Raatikainen, J-P. Skön, M. Johansson, U. Haverinen-Shaughnessy and M. Kolehmainen, Analysing School Buildings Energy Consumption Data Using Self-Organizing Maps, *Healthy Buildings*, 2012.

[12] J-P. Skön, M. Johansson, M. Raatikainen, U. Haverinen-Shaughnessy, P. Pasanen, K. Leiviskä and M. Kolehmainen, Analysing Events and Anomalies in Indoor Air Quality Using Self-Organizing Maps, *International Journal of Artificial Intelligence*, 9(12), 2012.

[13] M. Stocker, M. Rönkkö and M. Kolehmainen, Making Sense of Sensor Data Using Ontology: A Discussion for Residential Building Monitoring, *IFIP Advances in Information and Communication Technology, Artificial Intelligence Applications and Innovations*, 382, 2012, 341-350.

[14] J-P. Skön, O. Kauhanen and M. Kolehmainen, Energy Consumption and Air Quality Monitoring System, *Proc. of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2011, 163-167.

[15] J-P. Skön, M. Johansson, O. Kauhanen, M. Raatikainen, K. Leiviskä and M. Kolehmainen, Wireless Building Monitoring and Control System, *World Academy of Science, Engineering and technology*, 65, 2012, 706-711.

[16] M. Stocker, *Wavellite*, Available at: http://www.uef.fi/fi/envi/projects/wavellite, 2013.

[17] M. Neovius, *Trustworthy context dependency in ubiquitous systems*, (PhD thesis, Åbo Akademi University, Department of Information Technology, 2012).