Design Space Exploration for Secure Building Control

Martin Mansfield, Charles Morisset, Carl Gamble, John C. Mace, Ken Pierce, and John Fitzgerald

School of Computing, Newcastle University, UK
martin.mansfield@ncl.ac.uk

Abstract. By automation of their critical systems, modern buildings are becoming increasingly intelligent, but also increasingly vulnerable to both cyber and physical attacks. We propose that multi-models can be used not only to assess the security weaknesses of smart buildings, but also to optimise their control to be resilient to malicious use. The proposed approach makes use of the INTO-CPS toolchain to model both building systems and the behaviour of adversaries, and utilises design space exploration to analyse the impact of security on usability. By separation of standard control and security monitoring, the approach is suitable for both the design of new controllers and the improvement of legacy systems. A case study of a fan coil unit demonstrates how a controller can be augmented to be more secure, and how the trade-off between security and usability can be explored to find an optimal design. We propose that the suggested use of multi-models can aid building managers and security engineers to build systems which are both secure and user friendly.

Keywords: Security · Smart Buildings · Multi-Modelling · Design Space Exploration · Optimisation

1 Introduction

The critical services required to operate many existing buildings, such as Heating, Ventilation, Air-Conditioning (HVAC), lighting, access control and fire detection are automatically managed by building control systems. Reduced energy usage, improved efficiency, maintenance, comfort and safety are just a few of the many potential benefits automated building control can bring. In today's information age, these standalone building control systems are being connected to the Internet and other data networks to further improve operational efficiency and occupant safety by offering smart interactions, centralised and remote control, monitoring and maintenance. In doing so, smart buildings are becoming vulnerable to disruptive, damaging and life-threatening cyber-attacks (e.g. [1–3]). Furthermore, the increasing rate of connection to realise the potential operational benefits has exposed a lack of practical processes and mechanisms for securing existing building control systems.

Likely adversaries behind smart building cyber-attacks are varied: corporations, cyber-criminals, traditional criminals, occupants, nation states, hacktivists, and terrorists [4]. The different motivations, capabilities, resources and objectives these adversaries bring highlights the potential for building control systems to be attacked in many

different ways, some of which may not yet be evident. The best one can do is to identify a set of known feasible threats and defend against them by designing and implementing effective security mechanisms. A starting point is to consider building control systems as Cyber-Physical Systems (CPSs) due to their constituent cyber elements (e.g. controllers, adversaries) and physical elements (e.g. sensors, actuators, environment, users). Traditionally, cyber and physical security are treated as separate concerns, however the need to understand attacks and their impacts requires a holistic view incorporating both the cyber and the physical domains. Importantly, once secured, building control systems must still provide a required level of service which makes it necessary to also understand the impact security mechanisms would have on building operations before implementation (i.e. the security/usability trade-off).

In this paper we build on our multi-modelling approach introduced in [5], and show how practical security monitors for building control systems can be designed. We make use of the INTO-CPS open toolchain [6] to model building control systems and the attacking behaviour of potential adversaries towards those systems. We then utilise the Design Space Exploration (DSE) functionality of INTO-CPS to design practical security monitors by establishing the security/usability trade-off. The contributions of this paper are therefore as follows:

1. a security monitor model for a building control system cyber controller.
2. guidance for using DSE to explore the monitor's security/usability trade-off.
3. the extension of an existing case-study by including the security monitor.

To illustrate our modelling approach we consider the design of a security monitor for a single Fan Coil Unit (FCU), a common building control system found in buildings for managing room temperature. We explore just one aspect of the FCU's security by analysing a specific Man-in-the-Middle style attack launched against it. The attack illegitimately modifies sensor readings to maximise fan usage while minimising the deviation of temperature from the required set-point. An attack of this type could cause overheating and even fire, increase energy usage, wear and tear, routine maintenance and other financial costs. Building control systems including lighting accounts for 70% of a building's energy usage. If operated incorrectly (e.g. as a result of attack) a 20% increase in energy use can be expected in general [7].

Section 2 provides an overview of smart buildings and the INTO-CPS toolchain, as well as a summary of the existing use of multi-models to describe building control systems and security concepts. Section 3 introduces how controllers can be more resilient to attack by adding security monitors, and how DSE can be employed to establish a balance between secure and usable control. Section 4 demonstrates our approach by modification of a previous FCU case study, and Section 5 describes possible future directions for this work. Finally, concluding remarks are included in Section 6.

2 Background

This section introduces typical characteristics of smart buildings alongside their vulnerabilities. Additionally, it includes an introduction to multi-modelling and associated technologies, and how these technologies have been employed in the description of smart buildings and their potential attackers.

2.1 Smart Building Security

Building Control Systems Today's buildings take advantage of automated systems to control crucial operational services such as HVAC, lighting, water supplies, mobility, access control, and security, among others. The motivations for automating building control systems are improved operational efficiency, productivity, environmental sustainability, occupant health and safety, and reduced energy consumption. To achieve these potential advantages, building control systems integrate sensors to measure and collect environmental data from within a building (e.g. air temperature, humidity and occupancy). This data is transmitted to digital controllers which process it and calculate control instructions, which are then transmitted to actuators capable of altering the state of a building's environment. Lights may be turned on or off, air-vents opened or closed, a room's temperature raised or lowered, doors locked or unlocked, and so on. The integration of software-based cyber controllers and physical sensing and actuating devices means that most building control systems can be considered CPSs.

Security Concerns Traditionally, building control systems were standalone entities with no external connectivity. The security of these systems was provided largely by obscurity. Now, these once siloed systems are being retro-fitted with technologies connecting them to the Internet and other data networks to facilitate greater operational efficiency and safety. For instance, centralised monitoring and control, historical data storage, and remote maintenance (e.g. uploading software updates) can be supported. Consequently, the rapid hyper-connectivity of building control systems has opened up a large and complex cyber-attack surface and exposed the security provision for building control systems as being way behind the times.

One key reason for this is existing building control systems were built solely to work. Their designs typically came purely from civil engineering fields resulting in specifications that rarely stated the need for security. As a result, many of today's building control networks incorporate pre-existing legacy systems to which current security mechanisms cannot simply be 'bolted on' [2]. Also, memory, power and processing constraints of communicating devices (e.g. sensors) means security mechanisms such as data encryption are often too costly to implement. The use of common open protocols (e.g. BACnet [8] and KNX [9]) by disparate devices to facilitate communication exposes data traversing a network to varied cyber-attacks such as injecting fake sensor readings [3]. Furthermore, a lack of authentication processes enables attackers to use electronic devices (e.g. laptops, tablets and smart phones) to easily infiltrate and take control of systems on the network without detection.

The inherently insecure nature of building control systems exposes them to novel cyber-attacks such as disabling critical systems until a ransom is paid, or controlling systems to cause damage and disruption to the physical environment [1]. In the latter case, destructive commands could be transmitted over the building control network to place a system into a dangerous state for which it has not been designed. For instance, heating, air or water supplies can be disabled; rooms caused to overheat and damage sensitive data stores (e.g. patient records) or material (e.g. forensic evidence); systems overloaded to cause fires or floods while locking fire doors to trap occupants; or electronic doors unlocked to rooms holding sensitive information. Attackers have already

demonstrated they can launch highly damaging attacks on industrial control systems (e.g. explosions at a steel works [10] and nuclear power plant [11]). Attackers are now beginning to demonstrate they can exploit similar vulnerabilities in building control systems and gain control over them remotely [12, 13]. The inadequacies in building control network security means smart buildings are becoming an attractive target as they enable disruptive, damaging and life-threatening attacks with minimal effort.

2.2 Multi-modelling and Co-Simulation

Here, we give a brief introduction to the techniques used for modelling the case study, specifically a foundation of heterogeneous multi-modelling and design space exploration (DSE) techniques built on top of this.

INTO-CPS The INTO-CPS technologies[1] comprise a tool chain and supporting methods for model-based engineering of cyber-physical systems (CPSs) [14]. The core of INTO-CPS is support for definition and analysis heterogeneous system models, called *multi-models*, which combine individual models of the CPS' components and a description of their connections. The primary analysis technique for such multi-models is co-simulation, in which the individual component models are simulated together. The Co-simulation Orchestration E[ngine (COE) of INTO-CPS is called Maestro[2], which fully implements the emerging Functional Mock-up Interface (FMI) standard[3] for co-simulation. In FMI, component models are packaged into a standard format called a as Function Mock-up Unit (FMU). Maestro acts as a so-called master algorithm orchestrates the co-simulation, managing the passage of time and data exchange between FMUs. The INTO-CPS COE implements both a standard, fixed time-step algorithm and a variable time-step algorithm, which can speed up co-simulations and improve the fidelity of results for certain classes of FMUs. An INTO-CPS Association has been formed to continue work on the INTO-CPS technologies based around a community of industrial users. In addition to co-simulation, INTO-CPS provides support for automated testing, model checking, code generation, and hardware-in-loop (HiL) testing. Of relevance to the work in this paper are design space exploration (DSE), as described in the next section; and configuration of multi-model architectures through SysML, as described in Section 4.

The use of the FMI standard provides for an open tool chain that lowers the barriers to entry for model-based engineering, and allows for different paradigms of model to be connected together. This means that the most appropriate modelling formalism can be selected for each component of the system. Over 30 tools can produce FMUs, with more than 100 having partial or upcoming support[4]. At the time of writing (Q2 2018), Maestro is the most widely supported FMI engine, available on Windows, Linux, MacOS and ARM (Raspberry Pi). The case study described in Section 4 uses continuous-time (CT) model of the physical phenomena of the building system, combined with a

[1] http://into-cps.org/
[2] https://github.com/INTO-CPS-Association/maestro
[3] http://fmi-standard.org/
[4] http://fmi-standard.org/tools/

discrete-event (DE) model of the controller and security components. CT models represent systems as a set of differential equations which are solved numerically to provide high-fidelity simulations of physical phenomena, whereas DE models primarily represent data, state and events which alter these, and are best-suited for describing computing components. In this paper we use 20-sim for CT modelling, which describes models using graphs of connected blocks or icons [15], and VDM-RT for DE modelling. VDM-RT is an extension of the well-established notation VDM (Vienna Development Method) [16] that includes features required for description of real-time controllers including as classes, object orientation and native support for a model of computation time and distribution of functionality between compute units [17].

Design Space Exploration It is likely that there are many choices to be made when designing a CPS and these choices will affect the resulting performance of the CPS. Choices could include physical properties of the CPS, such as the thickness of walls in a building or the number or placement of sensors within a room, or they could regard cyber properties such as the choice of algorithm controlling heating or the frequency at which sensors are sampled. These choices along with the options for each define the design space for the CPS. One use for a multi-model then is to allow the engineer to explore the design space to find design options that are optimised with respect to one or more performance measures. Many of the design choices can be left open by the domain experts that produced the original models by exposing them as parameters of the resulting FMUs, in which case the user has the option to make use of the DSE facilities included in INTO-CPS to automatically explore the design space [18].

As a minimum, a DSE requires the definition of three aspects. The first aspect, parameters, is where we describe which parameters the DSE search may change and also gives a list of values each parameter may take. These parameters define the design space that is to be searched.

The second and third aspects relate to how we measure performance of a system and how we compare different designs using those measures. INTO-CPS simulations produce results in three forms, live graph plots of variables during a simulation, logs of monitored variables in CSV format and also 3D visualisations of the models if the user has created one. Since DSE is likely to run a great many simulations it is not practical for a user to observe all the live plots or 3D visualisations and so DSE makes use of Objective scripts that process the CSV simulation logs to produce objective values that characterise the performance of a CPS during simulation. Such objective functions might compute the total energy consumed by a system or the maximum deviation of a variable from an acceptable value. Once the objective values are computed for each design, they may then be used to compare different designs. If there is only a single performance measure then results may simply be placed in an list, ordered by that measure, to find the best, however, if there are multiple measures then a different means for comparison must be used. In the latter case, INTO-CPS makes uses the Pareto method to present the user with a non-dominated set of best designs [19].

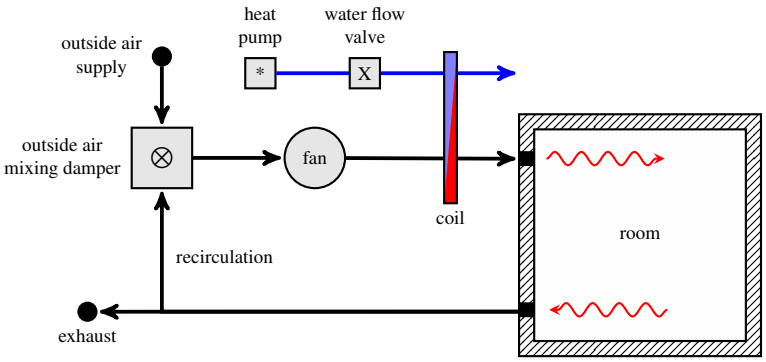

Fig. 1. Overview of the fan coil unit (FCU) example.

2.3 Modelling Building Control Systems

One cyber-physical building control system common to smart buildings is the Heating, Ventilation, and Air-Conditioning (HVAC) system. An HVAC system is responsible for controlling the air temperature and quality of a building, and does so using one or more Fan Coil Units (FCUs). Each FCU is comprised of several (physical) components for sensing and controlling temperature, and a (cyber) controller, implemented in software and responsible for the coordination of actuators based on sensed data. With a typical FCU able to service up to 150m^2, it is typical for a single building to include many FCUs.

An abstracted overview of an FCU is given in [20], and illustrated in Figure 1. The FCU uses a *fan*, to intake air and pass it across a cooling/heating *coil* and into a *room*. A bidirectional *heat pump* uses water to control the temperature of the coil, where the rate of temperature change is determined by the rate of water flow from the heat pump to the coil, controlled by a *water flow valve*. Both the fan speed and valve position are set by a digital controller. An *outside air supply* ensures adequate ventilation, and is mixed with *recycled air* recirculated from the room by a *outside air mixing damper*, with any excess leaving the system via an *exhaust*.

An initial FCU model proposed in [20] is transformed into a multi-model in [21]. The FCU multi-model contains 3 constituent models:

External A continuous-time model implemented in 20-Sim which specifies external stimuli, including a room air temperature set-point and the outside air temperature.

RoomHeating A continuous-time model implemented in 20-Sim comprised of two sub-models: `Room` and `Wall`. The `Room` model calculates the current room air temperature based on the fan speed, the water flow rate, and the wall surface temperature. The temperature of the wall surface is calculated by the `Wall`C model and is based on the outside air temperature and the room air temperature.

Controller A discrete-event model implemented in VDM-RT, this model calculates the fan speed and the aperture of the water flow valve based on the room air temperature and room air temperature set-point.

The model presented in this paper builds upon that presented in [21] by adding an explicit cyber-attacker model, an updated controller model that aims to address this attack, and modified objective scripts to capture the usage of the fan.

Modelling Adversaries The FCU multi-model is extended in [5] to demonstrate the use of multi-models in assessing the security of building control systems. The multi-model is extended by introducing an `Adversary` model which intercepts and potentially modifies the room air temperature set-point being communicated between the `Environment` and the `Controller`. The `Adversary` model is a discrete-event model implemented in VDM-RT.

By modifying the room air temperature set-point, the adversary executes a basic attack on the system by manipulating the controller in order to expedite wear and tear on the FCU by maximising fan usage. The implementation of this Man-in-the-Middle type attack continually increases and decreases the room air temperature set-point at a given frequency, causing the FCU fan speed to oscillate. Listing 1.1 outlines the control loop of the adversary model, which includes parameters for the frequency of room air temperature set-point modification (`attackFrequency`), and the upper (`upperModificationLimit`) and lower (`lowerModificationLimit`) limits of the modified room air temperature set-point.

```
instance variables
ACSP : real := 0.0 -- Attack Current Set-Point

operations
public setAttack: () ==> ()
setAttack()==
(
  let SP = RATSP_IN.getReading() in
  if SP > 0.0 then
    if ACSP < SP then ACSP = SP + upperModificationLimit
    else ACSP = SP - lowerModificationLimit;
  )
  else ACSP = SP;


  RATSP_OUT.setState(ACSP)
);

thread periodic(attackFrequency)(setAttack);
```

Listing 1.1. FCU Man-in-the-Middle attack where `RATSP_IN` is the intercepted room air temperature set-point and `RATSP_OUT` is the (potentially) modified room air temperature set-point sent to the controller.

3 Secure Controller Design

In this section we propose an approach to using INTO-CPS multi-models for the design of controller augmentations intended to make the control of smart building systems more secure. Furthermore, we introduce how DSE can be used to explore the trade-off between security and usability inherent in the design of such a controller, and inform the design of an optimised solution.

3.1 Security Conscious Control

In an attempt to negate the impact of harmful or malicious use of a smart building and its constituent systems, we propose the addition of a *security monitor*, which observes any input parameters and intercepts usage patterns which might cause damage. In monitoring inputs, the system might utilise a range of metrics to assess system use, such as the frequency of instructions sent to the controller, the rate of change of inputs, or the calculated ware on equipment.

It is important that a security monitor can be included in a diverse set of control systems, so its design should be modular and independent of any particular controller. By designing the controller to be added to a generic input stream, it can be made suitable for inclusion in a range of applications, including integration with legacy systems and those with which limited information about their operation is available.

By employing a multi-model based approach in the design of such a monitor, we can constrain its implementation to an independent model to effectively demonstrate how security monitoring can be added to existing systems without their modification, and that any necessary computation can be executed on separate hardware.

3.2 Controller Optimisation

As described previously, there are multiple system metrics that may be evaluated when modelling the FCU CPS, though perhaps not all are critical for the design of the controller and security modules. It is important then that the stakeholders of the system are consulted and the appropriate metrics are highlighted. As Avizienis *et al.*[22] tell us, security is

> *"...a composite of the attributes of confidentiality, integrity and availability..."*

thus we should pick metrics that speak to these and in this way we can observe the trade-off of these antagonistic concerns.

We consider the trade-off between integrity (security) and availability (usability) of a system. In designing the security monitor, a more restrictive approach to security is likely to restrict the usability of the system in some way. By employing DSE in the design in such a monitor, we are able to explore the impact of each design on both of these considerations, enabling the selection of a design which falls within a desired threshold for some metrics of both security and usability.

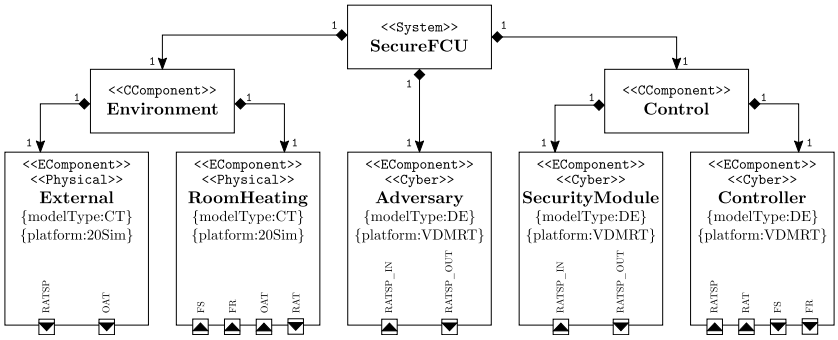

Fig. 2. Architectural Structure Diagram of secure FCU with adversary example.

4 Fan Coil Unit Case Study

In this section we illustrate our approach using a case study of an FCU. We describe an augmentation of the FCU multi-model provided in [5], which includes an additional model to undertake security monitoring. The multi-model is created using INTO-CPS technologies, and its simulation is used to determine fan usage in both the presence and absence of an attack. DSE is used to explore the trade-off between security and usability which results from varying the severity of response by the security monitor.

4.1 Security Controller Specification

We extend the FCU multi-model described in [5] by the addition of an additional `SecurityModule` model. The INTO-CPS SysML profile [23] is used to define the multi-model and its constituents in an Architectural Structure Diagram (ASD), illustrated in Figure 2.

The multi-model comprises 5 FMUs: `External`, `RoomHeating`, `Controller` and `Adversary` models are included from [5], and a complementary `SecurityModule` implements the behaviour of the security monitor. The `SecurityModule` model is a discrete-event model implemented in VDM-RT.

Each model is encapsulated in an independent FMU, and each FMU is defined using the *Encapsulating Component* (`<<EComponent>>`) stereotype. Additional parameters specify the model type (continuous/discrete) and a platform (in this case, VDM-RT and 20-Sim) for each FMU. The INTO-CPS SysML profile facilitates logical grouping of independent models by use of the *Collections Component* (`<<CComponent>>`) stereotype. This mechanism is used to indicate a relationship between `External` and `RoomHeating`, described as `Environment`, and similarly a relationship between `SecurityModule` and `Controller` described as `Control`.

The ASD is also used to define an interface for each FMU, specified as a series of ports which either input or output data to or from the model. Each port is labelled to describe the nature of the data it communicates, and an arrow specifies the direction of data flow. Data exchanged in the model includes the temperature of air both inside and outside of the room (*Room Air Temperature* (`RAT`), and *Outside Air Temperature*

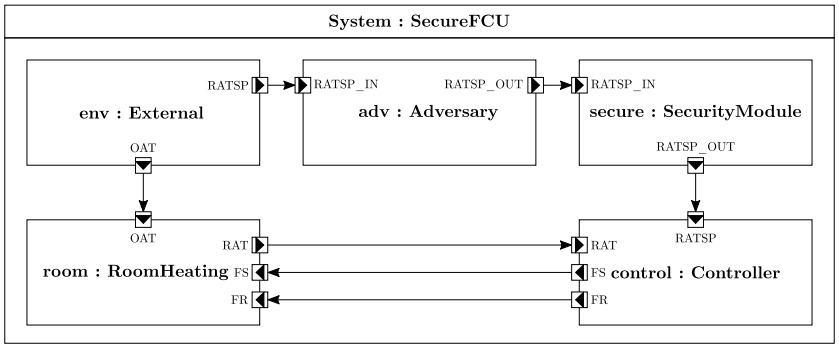

Fig. 3. Connections Diagram of secure FCU with adversary example.

(`OAT`)), the current desired temperature (*Room Air Temperature Set Point* (`RATSP`)), as well instructions for actuating temperature change (*Fan Speen* (`FS`) and *Flow Rate* (`FR`)). Data transfer between constituent models is defined using Connections Diagram (CD), illustrated in Figure 3, which makes connections between ports explicit.

The attack executed by the `Adversary` model accelerates wear on the FCU fan by instructing frequent temperature set-point changes to the system. To provide a countermeasure to this attack, the proposed security monitor filters fluctuating inputs using a moving average. By taking the average input over a sample of some period, the impact of input fluctuations can be significantly dampened, however this can introduce some delay in the FCU actuating the desired change in temperature. Listing 1.2 outlines the control loop of the security monitor model, which includes a parameter for specifying the length of the sample period (`sample_period`).

4.2 Controller Optimisation

To perform a DSE for optimisation, we need to detail both the range of the search and how results are to be computed. In the case of the FCU example we start by defining a range of values for the security module sampling period, here the range has a lower bound of one sample and an upper bound of 500 samples. The sample rate is one per minute and so the upper bound essentially averages the temperature set-points over an entire working day.

The metrics selected reflect two of the three security properties described by Avizienis *et al.*[22]. Specifically we evaluate the usage of the fan as an indicator of integrity, here the designer of the FCU has specified an acceptable fan usage limit of 25 and an attacker may attempt to push the usage beyond this limit. The second metric, temperature deviation, relates to the availability of the FCU for proper operation, i.e. achieving the desired room temperature. The acceptable range for temperature deviation is 3°c The best designs would minimise both of these metrics.

The DSE was performed under two sets of conditions (scenarios), the first is a normal working day, starting at 08:30 and finishing at 17:00, with the heating being off before 08:30, set to 21°c during the working hours and then off again after 17:00, this

```
instance variables
samples : seq of real;

operations
private monitorInput:()==>()
monitorInput()==
(
  if len samples = sample_period then samples := tl samples;
  samples := samples ^ [RATSP_IN.getReading()];
  RATSP_OUT.setState(sum(samples) / len samples);
);

functions
sum: seq of real -> real
sum(s) == if len s = 1 then hd s else hd s + sum(tl s);

thread periodic(monitorFrequency)(monitorInput);
```

Listing 1.2. Security monitor where RATSP_IN is the desired room air temperature set-point and RATSP_OUT is the filtered value sent to the controller.

is named 'without attacker' in the results. The second scenario uses the same working times and temperature set-points, but this time the cyber attacker is active and is able to intercept and change the room set-point as described earlier in Section 2.3.

The results of the search, in terms of the range of fan usage and temperature deviations for each controller frequency tested, are shown in Figures 4 & 5 respectively. On these graphs the colouration is used to indicate those controller frequencies that passed the constraint for that measure (coloured green) or that failed to meet the constraint (coloured grey). This is a departure from the normal INTO-CPS DSE result, where a Pareto analysis is used and colour represents the relative rank of a simulation result. Here the graphs indicate that there is a tension, with the best results for each measure being found at opposite ends of the controller frequency range.

Figure 6 presents a view of the results where the two objective measures are represented on the axis and the green colouration is only applied to designs that meet the constraints for both fan usage and temperature deviation, both with and without the attacker being active. Each '.' and '+' connected by a line represent a single design (controller frequency), with the location of the '.' and '+' indicating the fan usage and temperature deviations for that design. This view supports the stakeholders understanding what options are available in terms of the two measures both with and without the attacker, providing the stakeholders with information that allows them to explore the range of acceptable results to trade off between the two measures.

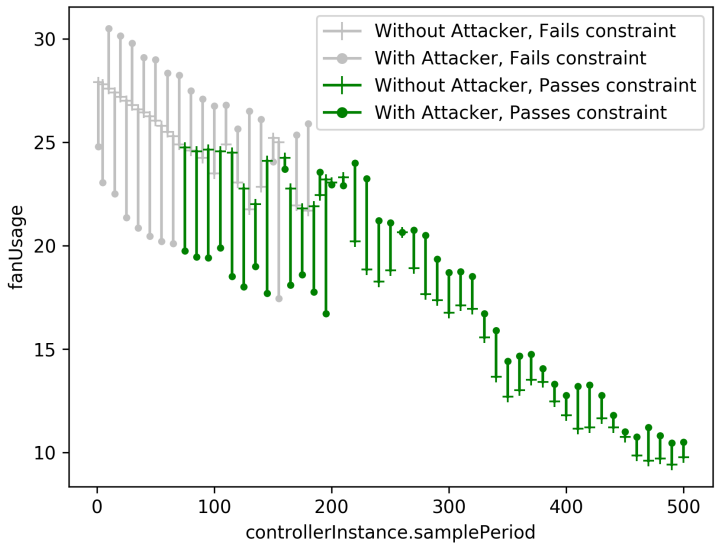

Fig. 4. Results showing the range of fan usage values for each controller frequency simulated.

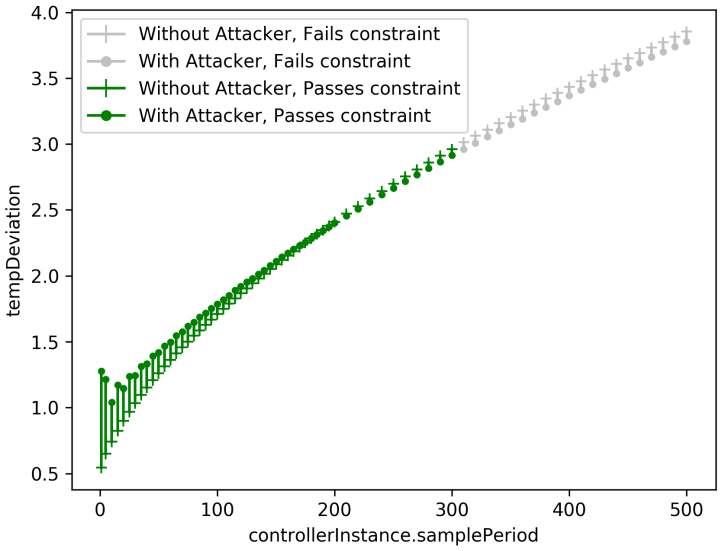

Fig. 5. Results showing the range of temperature deviation values for each controller frequency simulated.

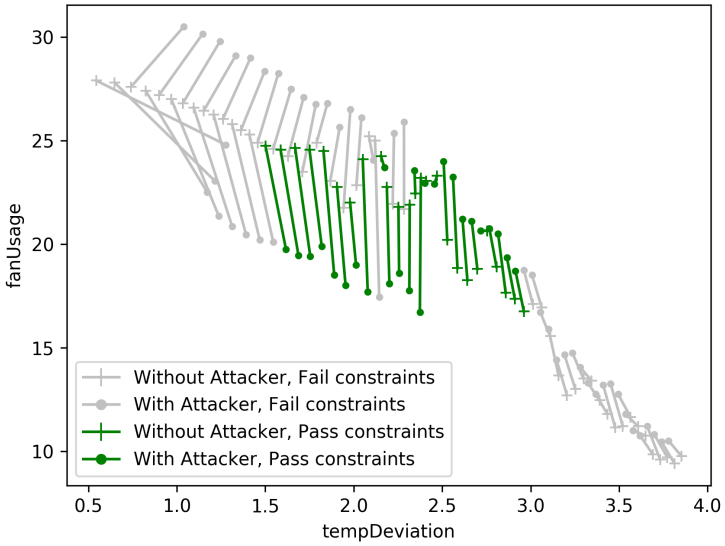

Fig. 6. Feasible security monitor designs within the bounds of maximum fan use of 25 and maximum temperature deviation of 3.

5 Future Work

The results presented in the previous section illustrate how DSE can be used to design a security controller, while assessing the security/usability tradeoff. It is particularly useful to be able to compare the effects on fan usage and temperature deviation with and without an attacker, since in practice, we cannot be certain whether an attacker is actually present or not.

The approach we presented in this paper is a stepping stone towards a generalised security controller design method, where it should be possible to sweep through multiple scenarios. Indeed, different users might have different usability requirements or ways of behaving within the system, and the security controller should be designed accordingly. However, this is likely to require the usage of probabilistic or statistical models, which are not yet supported in INTO-CPS.

The graphs in Figures 4 & 6 show that for fan usage, the range of controller frequencies that pass the constraint is not continuous. This is indicated by the presence of grey, non-compliant results, within the block of green, compliant, results. This emergent behaviour is believed to be due to the relative frequencies of both the controller and the attacker and indicates that the optimal controller frequency could differ with changing attacker frequencies. Thus we will expand upon the DSE reported in this paper to alter both controller and attacker parameters in the same search, to better understand the relationship between them.

Similarly, we would like to explore the DSE of the attacker and of the security controller using game theory, as it is likely that, in practice, the attacker will adapt its behaviour to that of the security controller, and conversely. Ideally, the tool support should help looking for a Nash equilibrium (i.e., for a configuration where neither the attacker nor the security controller has any incentive in changing their strategy).

Finally, we would also like to explore the design of more complex attackers, for instance multiple attackers synchronising their attacks, or attackers learning the behaviour from the users to increase their impact.

6 Conclusions

In this paper, we present an approach to the utilisation of multi-models in the design of a security controller for cyber-physical systems, particularly for the control of smart building systems. We have demonstrated how DSE can be particularly helpful in exploring the inherent trade-off between security and usability considerations, and have illustrated our approach on a simple case study, by the design a security monitor for a FCU, where the attacker aims at over-using the fan by tempering with the temperature set-points, and the security monitor provides a counter-measure by taking a moving average over input values.

We believe our approach is a stepping-stone towards a more integrated method to assess and design security mechanisms for the control of CPSs, and opens the door for modelling experts to include more complex defensive and offensive mechanisms in the discrete models of both controllers and potential attackers, respectively.

References

1. H. Boyes, "Security, privacy, and the built environment," *IT Professional*, vol. 17, no. 3, pp. 25–31, 2015.
2. S. Mansfield-Devine, "The dangers lurking in smart buildings," *Computer Fraud & Security*, vol. 2015, no. 11, pp. 15 – 18, 2015.
3. T. Mundt and P. Wickboldt, "Security in building automation systems - a first analysis," in *Proc. of the International Conference on Cyber Security And Protection Of Digital Services*, Cyber Security, pp. 1–8, 2016.
4. ENISA, "Threat landscape for smart home and media convergence," 2015.
5. J. Mace, C. Morisset, K. Pierce, C. Gamble, C. Maple, and J. Fitzgerald, "A multi-modelling based approach to assessing the security of smart buildings," in *Proc. of the PETRAS, IoTUK & IET 1st Int. Conf. on Living in the Internet of Things*, 2018. In press.
6. P. G. Larsen *et al.*, "Integrated tool chain for model-based design of cyber-physical systems: The INTO-CPS project," in *Proc. of the 2nd Int. Workshop on Modelling, Analysis, and Control of Complex CPS*, pp. 1–6, 2016.
7. K. W. Roth, D. Westphalen, J. Dieckmann, S. D. Hamilton, and W. Goetzler, "Energy consumption characteristics of commercial building hvac systems volume iii: Energy savings potential," 2002.
8. S. T. Bushby and H. M. Newman, "BACnet Today: Significant new features and future enhancements," *ASHRAE Journal*, vol. 44, no. 10, pp. 10–17, 2002.

9. M. Ruta, F. Scioscia, E. D. Sciascio, and G. Loseto, "Semantic-based enhancement of ISO/IEC 14543-3 EIB/KNX standard for building automation," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 731–739, 2011.
10. "Hack attack causes 'massive damage' at steel works." http://www.bbc.co.uk/news/technology-30575104.
11. R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.
12. "Let's get cyberphysical: Internet attack shuts off the heat in Finland." https://securityledger.com/2016/11/lets-get-cyberphysical-ddos-attack-halts-heating-in-finland/. Accessed: 12-03-2018.
13. "Lock out: The Austrian hotel that was hacked four times." http://www.bbc.co.uk/news/business-42352326. Accessed: 12-03-2018.
14. P. G. Larsen, J. Fitzgerald, J. Woodcock, P. Fritzson, J. Brauer, C. Kleijn, T. Lecomte, M. Pfeil, O. Green, S. Basagiannis, and A. Sadovykh, "Integrated tool chain for model-based design of cyber-physical systems: The into-cps project," in *2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data)*, (Vienna, Austria), IEEE, April 2016. http://ieeexplore.ieee.org/document/7496424/.
15. C. Kleijn, "Modelling and Simulation of Fluid Power Systems with 20-sim," *Intl. Journal of Fluid Power*, vol. 7, November 2006.
16. P. G. Larsen, K. Lausdahl, N. Battle, J. Fitzgerald, S. Wolff, S. Sahara, M. Verhoef, P. W. V. Tran-Jørgensen, and T. Oda, "VDM-10 Language Manual," Tech. Rep. TR-001, The Overture Initiative, www.overturetool.org, April 2013.
17. M. Verhoef and P. G. Larsen, "Enhancing VDM++ for Modeling Distributed Embedded Real-time Systems," Tech. Rep. (to appear), Radboud University Nijmegen, March 2006. A preliminary version of this report is available on-line at http://www.cs.ru.nl/ marcelv/vdm/.
18. C. Gamble, "Comprehensive DSE Support," tech. rep., INTO-CPS Deliverable, D5.3e, December 2017.
19. J. Fitzgerald, C. Gamble, R. Payne, and B. Lam, "Exploring the cyber-physical design space," in *INCOSE Int. Symp.*, vol. 27, pp. 371–385, 2017.
20. J. Fitzgerald, C. Gamble, R. Payne, P. G. Larsen, S. Basagiannis, and A. E.-D. Mady, "Collaborative model-based systems engineering for cyber-physical systems, with a building automation case study," *INCOSE Int. Symp.*, vol. 26, no. 1, pp. 817–832, 2016.
21. M. Mansfield, C. Gamble, K. Pierce, J. Fitzgerald, S. Foster, C. Thule, and R. Nilsson, "Examples Compendium 3," tech. rep., INTO-CPS Deliverable, D3.6, December 2017.
22. A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 11–33, Jan 2004.
23. E. Brosse, "SysML and FMI in INTO-CPS," tech. rep., INTO-CPS Deliverable, D4.3c, December 2017.