Modular Safety Cases for the Assurance of Industry 4.0

Omar Jaradat[1]∗, Irfan Sljivo†, Richard Hawkins‡, Ibrahim Habli‡

∗National Electric Vehicle Sweden (NEVS) AB, Trollhattan, Sweden,

†Malardalen Real-Time Research Centre, Malardalen University, Vasteras, Sweden,

‡Department of Computer Science, University of York, York, UK

Abstract *The Internet-of-Things (IoT) has enabled Industry 4.0 as a new manufacturing paradigm. The envisioned future of Industry 4.0 and Smart Factories is to be highly configurable and composed mainly of the 'Things' that are expected to come with some, often partial, assurance guarantees. However, many factories are categorised as safety-critical, e.g. due to the use of heavy machinery or hazardous substances. As such, some of the guarantees provided by the 'Things', e.g. related to performance and availability, are deemed as necessary in order to ensure the safety of the manufacturing processes and the resulting products. In this paper, we explore key safety challenges posed by Industry 4.0 and identify the characteristics that its safety assurance should exhibit. We propose a modular safety assurance model by combination of the different actor responsibilities, e.g. system integrators, cloud service providers and "Things" suppliers. Besides the desirable modularity of such a safety assurance approach, our model provides a basis for cooperative, on-demand and continuous reasoning in order to address the reconfigurable nature of Industry 4.0 architectures and services. We illustrate our approach based on a smart factory use case.*

1 Introduction

The Internet-of-Things (IoT) can be seen as a system of inter-connected cyber-physical objects that collect and exchange data. More formally, IoT is defined as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving

[1] omar.jaradat@nevs.com, irfan.sljivo@mdh.se,
{richard.hawkins, ibrahim.habli}@york.ac.uk

interoperable information and communication technologies" [25]. This infrastructure allows the Things to be sensed and controlled remotely so that their integration into the physical world leads to different ways to utilise the Things in various reconfigurable applications. Cloud Computing is a fundamental infrastructural element for IoT, enabling different types of X as a Service (XaaS)[1] [19], where X is a software, platform, infrastructure, etc. In this paper, we adopt the NIST definition of Cloud Computing:

> "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*" [17].

The marriage of the IoT and Cloud services (e.g., cloud XaaS) has paved the way towards the fourth industrial generation[2], Industry 4.0, as a new trend of automation and data exchange in the manufacturing industry. This new industrial paradigm is characterised by its ability to reconfigure and often optimise autonomously, particularly during the operational stages. Moving certain manufacturing services, e.g. scheduling and data storage and analytics, to the Cloud has potential benefits in cost reduction, energy efficiency, sharing of resources and increased flexibility. The use of Cloud Computing in critical applications has been highlighted as a significant area of research, especially for production and manufacturing systems [3], [7], [12], [26].

However, factories are often categorised as safety-critical systems as failures of these systems, under certain conditions, can lead to human harm or damage to property or the environment, e.g. due to the use of heavy machinery or hazardous substances. As such, the risk associated with the manufacturing processes and the resulting products has to be analysed, controlled and monitored. However, the reconfigurable, modular and dynamic nature of Smart Factories pose significant safety assurance challenges. For example, designers or operators of factories do not have much control over the design and evolution of the 'Things' or Cloud-based services that are increasingly being used in manufacturing processes. This potentially weakens confidence in the safety of the factory and can undermine the overall safety case [21], i.e. due to high degrees of uncertainty about the actual performance or behaviour of these 'Things' or Cloud-based services.

Most of the reviewed published literature on IoT and Cloud Computing reveals focus on security in particular and dependability in general but without much focus on safety. For example, the German automation technology supplier 'PILZ' [18] stated that the Industry 4.0 vision entails modular plants being reconfigured quickly and flexibly. They view the control and decision-making process

[1] Key IoT terms are described in the last section.

[2] aka Industrie 4.0

in Industry 4.0 becoming more decentralised and highlight safety, in particular, as a fundamental challenge, with emphasis on the necessary modular certification of the individual factory devices (PILZ uses the term Safety 4.0 to indicate modular safety solutions).

In this paper, we introduce a common Industry 4.0 architectural style (Section 2) and explore its safety assurance characteristics (Section 3). We then propose a modular safety assurance model by diffusion of the different actor responsibilities, e.g. system integrators, cloud service providers and 'Things' suppliers (Section 4). Our model aims to provide a basis for cooperative, on-demand and continuous safety reasoning in order to address the reconfigurable and compositional nature of Industry 4.0 architectures. We illustrate our approach based on a smart factory use case (Section 5) and conclude in Section 6.

2 Industry 4.0 Architecture

In this section, we introduce a generic architecture for Industry 4.0. This architecture comprises three levels, as depicted in figure 1, where the Things and Fog/Edge levels typically represent the local part of the system, while the Cloud represents a remote infrastructure that is usually owned by a third-party service provider:

- *The Things Level* is composed of a set of Things that enable interaction with the physical environment via different sensing/actuating devices. We consider a Thing as an object capable of communicating with other networked devices [2]. Due to the limited storage and processing power, devices from this level rely on the Fog or Cloud infrastructures for storage and processing services.
- *The Fog Level* is composed of a set of Fog/Edge devices that are directly connected to Things or/and Cloud infrastructure. We consider Fog devices to be local computational devices that offer advanced storage and processing power to the Things and rely on remote Cloud infrastructure for high-power computing and storage. The Fog devices receive data from the Things and, depending on the system configuration, might forward the data to the Cloud infrastructure. Moreover, the Fog devices may perform partial processing of the data and directly instruct commands to the Things.
- *The Cloud Level* is composed of a set of remote servers providing on-demand capabilities. The Cloud infrastructure typically receives data from the Fog devices, processes the data and forwards commands to the Things via Fog devices.

The distribution of control, authority and responsibility between the Things and the Fog and Cloud infrastructures depends on factors such as (1) performance,

e.g. avoiding the Cloud for hard real-time requirements, (2) global and adaptive services, e.g. Big Data analytics via the Cloud and (3) local situational awareness, e.g. via smart IoT-based devices. Understanding the behaviour and integrity of the individual Things and infrastructural elements, and their interactions, is a pre-requisite for assuring the safety of Industry 4.0.

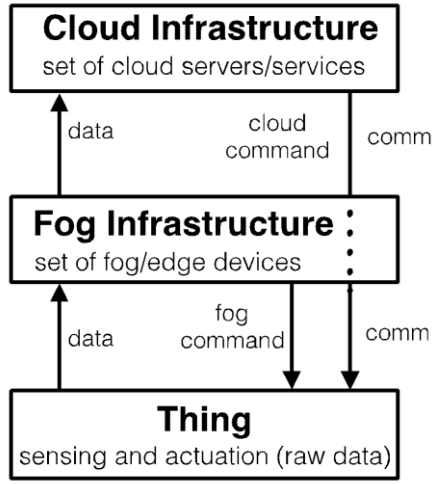

Fig. 1. Industry 4.0 Generic Architecture

3 Safety Characteristics for Industry 4.0

Considering the capabilities of Industry 4.0, in this Section, we explore key characteristics of its safety assurance.

1) *Modular and Cooperative:* The safety assurance for Industry 4.0 will often have to be cooperative in a sense that a safety or assurance case cannot be built by a single stakeholder or organisation. Since the implementation of the business models is shifting from a single company to a network of service providers [14], so does the resulting system shift from a standalone system to a network of devices and services, performing, cooperatively, a number of functionalities. Each business participating in the integrated system, e.g. as a Thing supplier (be it a "dumb" or a "smart" connected device), should accompany the provided Thing with a set of safety assurances for different usages. However, since the suppliers cannot provide all the needed safety assurances out-of-context, certain properties should be assured by the integrator in the context of the particular usage of the Thing.

2) *Continuous:* Safety cases are used to justify how the risk of each identified

hazard has been eliminated or adequately mitigated. Industry 4.0 assumes that a modular factory can be reconfigured quickly and flexibly. The safety assurance of such a factory is expected to be in a position to accommodate this widening of flexibility. For safety cases, they should comprise evidence to make a convincing argument to support the relevant safety claims [15]. However, some claims and pieces of evidence might get invalidated due to reconfigurations that commonly take place in the factory, e.g. changes to the manufacturing processes and services. Hence, safety cases might be out of date and no longer reflect the actual safety performance of the system. To this end, the safety cases should be proactively reviewed and continuously maintained in order to justify the evolving status of the factory [6].

3) *On-demand:* As motivated in the previous characteristic, safety cases should be maintained after changing the associated factory to continuously demonstrate the status of the safety performance. Sometimes, however, updating the safety cases is not feasible because of the nature of the changes. That is, there might be drastic changes to the factory that could introduce new and different types of hazards that require repeating the entire safety assurance process and generating more and/or new pieces of evidence. Here, re-constructing the safety cases might be necessary as a more cost-effective option compared to updating the existing cases [22].

In this paper, we limit our focus to the modular and cooperative characteristics of safety assurance for Industry 4.0, considering the overall safety case for Smart Factories and future needs for continuous and on-demand assurance.

4 Industry 4.0 Safety Assurance Approach

Assurance can be defined as justified confidence in a property of interest. In high-risk domains, assurance is typically demonstrated through the provision of an assurance case, consisting of a structured argument, i.e. justification, supported by evidence [15]. In this paper, the assurance case is for safety properties (aka safety case). As discussed in Section 3, due to the co-operative nature of IoT, it is not possible for any single stakeholder to provide the assurance case for the entire system.

The constituent Things, and the required infrastructure elements will be developed and provided by different organisations. It is these separate organisations that have the knowledge of the properties and characteristics of their components (i.e. Things or infrastructure elements). However, these suppliers are only able to reason about the assurance of their own components and can say little about the assurance of the IoT system as a whole, especially with regard to system- level conditions such as hazards, accidents and harm. The system integrator must

therefore consider what is required for safety assurance and then show that the Things or infrastructure elements being used are able to support this.

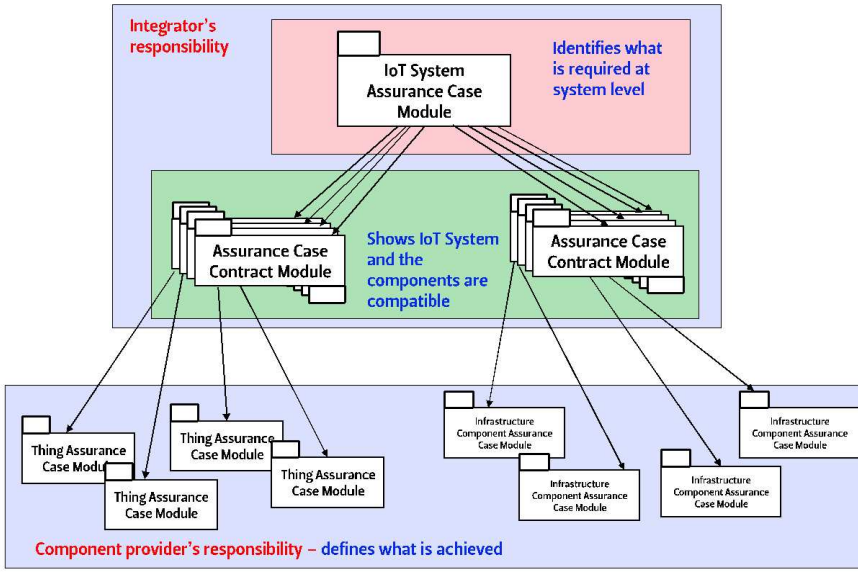

Fig. 2. Proposed IoT Assurance Case Architecture

This leads us to propose a modular approach to assurance for IoT-based systems as indicated in figure 2. The figure shows the overall assurance case structure for the IoT-based system, split into a number of modules, where each module reasons about a different aspect of the system. There are assurance modules for each of the Things and infrastructure elements, and modules dealing with the assurance of the integration of these into an IoT system. The different stakeholders have assurance responsibilities within the structure in figure 2 in order to ensure that a compelling overall assurance case for the IoT system can be created. These responsibilities are discussed below.

4.1 Responsibilities of Things or Infrastructure Providers

Each of these providers must define an *assurance contract*. This contract defines the set of properties that they are able to assure and a definition of potential failure behaviour of their Things or infrastructures. In order to be usable as part of the integrated assurance case for the IoT system, each of the identified properties should be defined with the following assume-guarantee reasoning form:

if {condition} then {Thing or infrastructure} shall provide {property} with confidence of {confidence}

The *condition* and *property* represent the assumptions and guarantees of an assume-guarantee contract [23]. The *condition* and *confidence* of this assume-guarantee contract specification is crucial to our approach. For any *Thing* or *infrastructure*, there exist limitations on the circumstances under which it can perform its function. For example, an assurance contract for a pressure sensor may include:

If temperature is greater than -20°C then pressure sensor shall provide air pressure value with accuracy of 0.001% with confidence of 99%.

It should be noted that, unless some failure has occurred, the pressure sensor is expected to provide an air pressure value. However, at temperatures below -20°C the confidence in that value will be reduced. If this confidence is not defined at these lower temperatures, then the property cannot be assured outside that temperature range. This may then require alternative pressure sensing capabilities (or some other guarantee of temperature range) in order to create the assurance case.

Knowing the level of confidence with which a Thing or infrastructure can guarantee a particular property is also crucial to the integration process as it enables the overall level of assurance for the system properties to be determined. Further, each Thing or infrastructure provider must be able to reason about the completeness and correctness of the failure behaviour definition provided as part of the contract. These definitions of such failure behaviour are also taken into account when assessing the assurance of the integrated system. It should be noted that the information required of the Thing or infrastructure provider described above is specific to the Thing or infrastructure, but in no way specific to the particular IoT system of which that Thing or infrastructure may become a part. This facilitates the use of independently, commercially developed and reusable components as part of the safety assurance framework.

4.2 Integrator's Responsibilities

The integrator has responsibility for creating the IoT system by utilising the Internet-enabled Things and infrastructure elements. The integrator therefore also has responsibility for demonstrating the overall safety assurance of the IoT system. As previously discussed, the integrator should have available to them information about the assurance of the individual Things or infrastructures through the assume-guarantee contract specifications. The integrator must show how the assurance provided for the Things or infrastructures can be used to demonstrate

the assurance of system- level properties. In particular, the integrator must identify the hazards, i.e. sources of potential harm, and their associated risks, posed by the system, e.g. unsecured loads, laser radiation or heavy machines operating in the presence of operators. For any configuration of Things or infrastructures, the integrator must then determine the safety requirements for each of these by identifying how the Things or infrastructures may contribute to hazards (this could for example be done through considering deviations on the functionality or interactions).

Once these requirements are known, the safety assurance case for the IoT system can be created if it can be demonstrated that 1) the properties in the contracts are able to satisfy the assurance requirements defined for the IoT-based system with sufficient confidence, and 2) the contracts of the relevant Things or infrastructure elements are satisfied (the properties and conditions are met and the failure modes are mitigated). As discussed, the Thing or infrastructure element provider has responsibility for specifying the contract for that element and ensuring the properties are met, however it is the responsibility of the integrator to ensure the conditions are satisfied, and the identified failure modes of the element are mitigated in the context of the overall IoT system (through a variety of mechanisms such as redundancy, monitoring, operational constraints etc.).

In order to facilitate this integration of an overall safety case, we propose the use of *assurance case contracts*. Assurance case contracts provide a mechanism for recording and justifying the agreed relationship between assurance case modules. Figure 2 shows assurance case contracts being established between the IoT-based system assurance case module and the individual component modules. The structure that such a contract module might have is illustrated as a pattern in figure 3, using the Goal Structuring Notation (GSN). Readers who are unfamiliar with this notation are referred to the GSN Standard [1] for more detailed information.

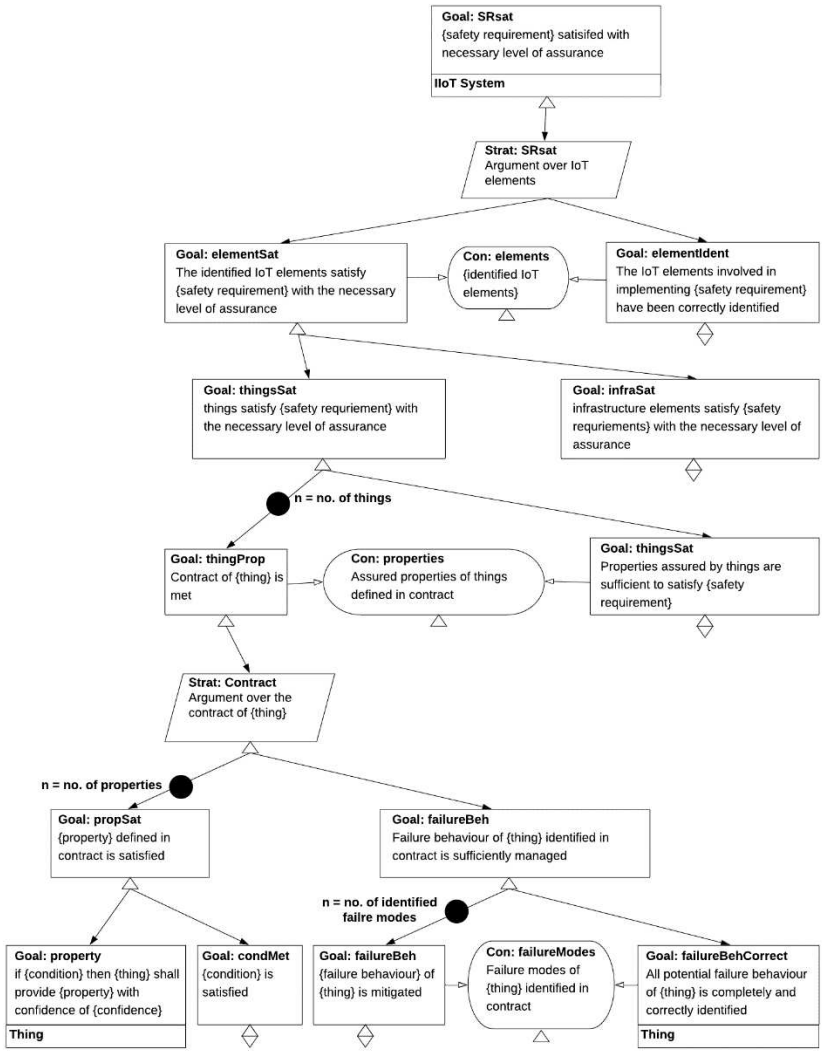

Fig. 3. Structure of an Assurance Case Contract for IoT System

Figure 3 shows how in order to assure a safety requirement identified by the integrator, a number of the Things or infrastructure elements may need to be considered. For each of these, the contract defined for those elements is used to make the assurance argument. In Figure 3 we show only how this is done for Things, but a similar argument structure would be used for infrastructure elements as well. In order to form the assurance case contract, it must be demonstrated that the properties defined in the contract for each element are sufficient to satisfy the

safety requirement. It must then be demonstrated that each aspect of the contract for each element is satisfied. Claims about the satisfaction of the properties, and the identification of failure behaviour, are supported by a safety case module developed by the provider of that element and provided to the integrator along with the element itself.

Needless to say, establishing and justifying assurance case contracts is a challenging task. The specification of the assurance model and clear definition of the supplier's assurance responsibilities are merely a first step towards this. A contract- based assurance approach is potentially desirable for an IoT- based system as the contract helps to determine whether the relationship between the assurance case modules continues to hold and the (combined) safety assurance case remains valid when Things or infrastructures are altered or substituted in the system. This issue is discussed further in Section 6.

5 Use Case

In this section we present a fictitious, yet representative, Smart Factory and focus on a single part of the factory to illustrate safety assurance for Industry 4.0. We focus on a Warning Light System (WLS) as a safety measure that includes IoT-related elements. We demonstrate our approach by performing safety analysis of the WLS and developing a corresponding argument for the system based on the assurance case contract structure presented in Section 4.

5.1 Smart Factory Description

Our use case considers scenarios where the requirements and design specification for the manufacturing of a product are provided via a Cloud-based service. Some of the manufacturing control capabilities reside remotely on the Cloud, e.g. scheduling and design reconfiguration. Others are managed locally either at the Fog or Things levels. More specifically, our use case considers a manufacturing factory in which a number of computer-based machine tools make a range of gearbox shafts from metal blanks. The blanks, which weigh about 4kg each, are delivered in pallets of 50, and stored in an automated warehouse until they are required. Finished products are also packed into pallets and taken to a holding area before being shipped to the main assembly plant.

The movement of pallets around the plant is managed using an Automatic Guided Vehicle (AGV) system. The system consists of a number of battery-powered vehicles, each fitted with pallet handling equipment, whose movements are

directed by an AGV Central Control Fog. This is interfaced to a Warehouse Control, Holding Area Control and Machining Control Fogs, so that stock movement requirements can be fulfilled. Each AGV will carry only one pallet at a time. The conceptual flow of materials is illustrated in figure 4.

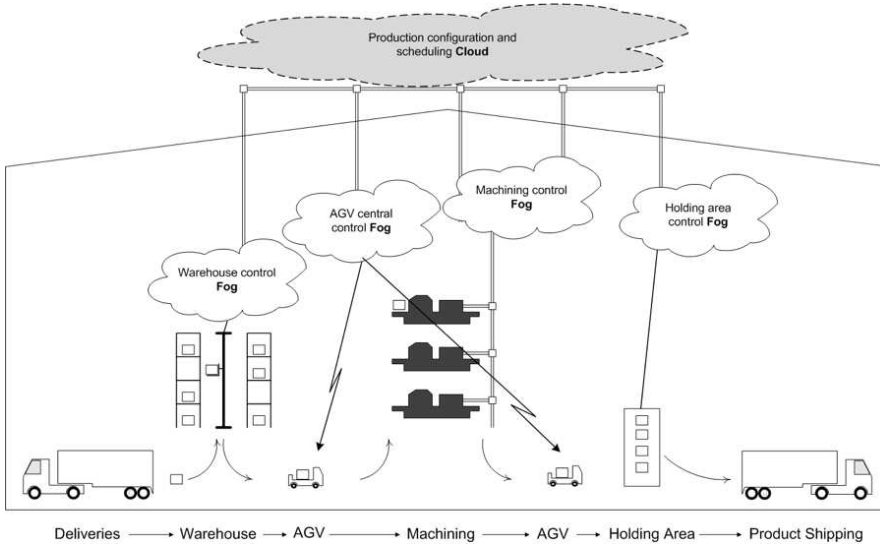

Fig. 4. Flow of Materials and Information through the Factory

To manage different automated activities in the factory, Light Imaging, Detection, and Ranging (LIDAR) sensors are positioned to cover the whole factory. Such a setup allows the Smart Factory to "see" what is going on, i.e. in real time, and to manage the activities accordingly. A Cloud service is used for the integration of the LIDAR inputs and for modelling the activities in the factory. This Cloud service allows for customisable features to be implemented specific to different factory operations. Special docking stations are provided for the AGVs, each weighing about 0.8 tonnes. The vehicles will normally be directed by the central Fog to return to these charging stations when they are not required to move pallets. The factory is not fully automated, and people cannot be excluded from the areas where the AGVs operate.

5.2 Hazard Analysis

Since the factory employs both human workers and machines of different autonomy levels, there are many factory-level hazards, e.g. proximity to heavy moving objects. One general safety measure is to define restricted areas for the different

factory configurations to protect the human workers from both the moving machinery and the dangerous goods they transport. In this use case, we focus on a single factory-level hazard: *"Unauthorised AGV vehicle enters the restricted area"*. Due to the noise protection procedures that human workers may be using in certain configurations, audio warning is not sufficient, so a visual warning light system is also needed. Amongst the different safety requirements specified to address this hazard, we focus on the following requirement: *"A warning light shall be signalled when an unauthorised AGV enters the restricted area"*. This requirement is allocated a Safety Integrity Level (SIL) 2, based on the likelihood and severity of the considered factory-level hazard.

To achieve this requirement, several other sub-requirements should be specified. We mention only some:

R1: The system shall distinguish between authorised and unauthorised AGVs.
R2: The scope of the restricted area shall be specified to 5cm degree of precision.
R3: The signalling of the warning light shall occur within 0.5sec from an unauthorised AGV entering the restricted area.

The main objective of the proposed Warning Light System (WLS) is to monitor restricted areas where certain types of objects (humans, robots, vehicles, etc.) are prohibited due to safety reasons. The system is intended to trigger a warning light if an object classified as prohibited under the given factory configuration appears in the designated restricted area.

The high-level architecture of the WLS is presented in figure 5. WLS is implemented using the Cloud service and the factory LIDARs. We focus on a particular configuration and a specific restricted area for that configuration, as presented in figure 5. The considered restricted area includes 3 LIDARs, 4 access points and 4 warning light lamps. The gateway and local control device are located within the factory, but outside of the restricted area. The access points facilitate wireless communication between the sensors/lamps and the gateway, while the gateway enables connection to the cloud service that acts as a control node.

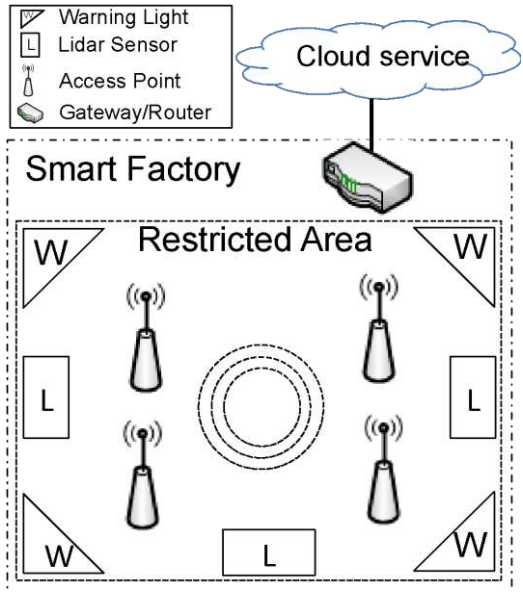

Fig. 5. Smart Factory Use Case: WLS Configuration

The considered WLS is composed of:

Things:

1) LIDARs 1-3 (identical)
2) Warning Light 1-4 (identical)

Infrastructure elements:

3) Gateway/Router (local control node)
4) Access Point 1-4 (identical)
5) Cloud Service (control node)

The Cloud service is responsible for processing the data and commanding the activation of the warning light via the local network. The Cloud is also responsible for monitoring all moving objects in the factory. The local controller is only responsible for the most severe restricted area violations. As such, it only monitors certain objects entering the area. The initial requirements are further decomposed and allocated to the IoT system elements to more clearly specify their function. For example, for the R1 requirement we specify sub-requirements such as:

R1.1: LIDARs shall detect all objects entering the restricted area.
R1.2: The cloud service shall analyse and classify all detected objects.
R1.3: The gateway shall analyse and classify only the most dangerous objects.
R1.4: The gateway shall transmit all the sensor data to the cloud service.

Similarly, for the requirement R3, we decompose it to allocate the timing requirements on the operations of the different elements. For example, a sub-requirement R3.1 can be specified as: "The warning lights shall engage on receipt of the engage command within 0.2sec".

5.3 WLS Failure Analysis

So far, we have defined safety requirements for WLS without considering failures of the individual elements. In this section we consider hazardous contributions of all the WLS IoT system elements and their contributions to the considered hazard. Some of the identified hazardous failures for the IoT system elements are as follows:

- LIDARs
    - No signal provided
    - Unable to detect unauthorised object entering restricted area
    - Signal reports incorrect light conditions
- Warning light lamps
    - The warning light does not turn on when requested
    - The warning light turns on with a delay greater than 0.2sec
- Access Points
    - Access point fails to route data to the Gateway
    - Access point takes longer than intended to route data
- Cloud Service
    - Cloud does not generate warning signal request
    - Cloud generates an incorrect warning signal request
    - Cloud takes longer than intended to generate warning signal request.

We have also derived safety requirements to address the above hazardous failures. For example, these requirements include the following:

1) "Each restricted area shall have at least two warning lamps visible from every position in the area",
2) "Each moving object in the factory shall have a marking detectable by LIDARS", and

3) "Human workers shall be notified of the WLS failures".

All the derived requirements are assigned with at least SIL 2, based on the corresponding higher-level requirement.

5.4 Assurance Case Contract Example for WLS

The application of the assurance case contract, as defined in Section 4, is presented in figure 6.

In the presented argument we focus on the safety requirement R3 of WLS and detail in particular the warning light lamp element. The supplier of the lamp is able to provide an assurance case for the lamp that supports various claims about the lamp as detailed in the assume-guarantee contract. In the example in figure 6 we see that the lamp assumes a constant power supply and working temperature in a predefined range in order to provide assurance of maximum light intensity within 0.2 seconds during the promised lifespan.

The confidence in this claim is provided by the lamp assurance case. In forming the assurance case contract shown in figure 6, this claim about the lamp is used to support a safety requirement as part of the higher-level factory assurance case (in other words, the assurance case contract reasons that this lamp is sufficient, from a safety perspective, for its use as part the factory operations).

Figure 6 shows how the assurance case contract also must consider the known failure behaviours of the lamp as detailed by the supplier. The effects of the failure behaviours are shown to be mitigated by the AGV and the Smart Factory configuration.

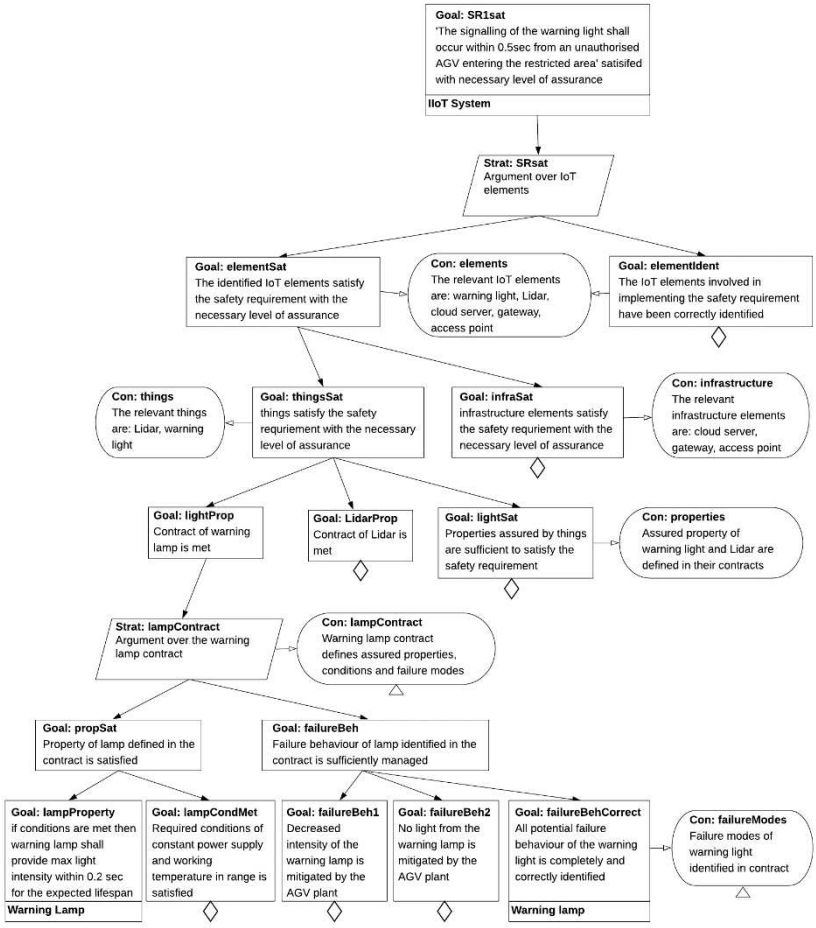

Fig. 6. The Warning Light Lamp Assurance Case Contract

6 Discussion and Conclusions

We have highlighted a number of safety challenges posed by Industry 4.0 and proposed a modular assurance approach that has the potential to address some of these challenges, particularly with regard to the compositional and configurable nature of IoT-based architectures. In essence, our approach builds on past and current research on assume-guarantee reasoning, contract-based assurance and modular certification for safety-critical applications [20] [16] [8]. Historically, these approaches formed the basis for safety cases and certification for systems

in various domains including automotive [24] and aviation [5]. However, some fundamental safety assurance problems remain and have to be addressed as a prerequisite for realising the general-purpose vision of Industry 4.0. We explore, and reflect on, these in the rest of this section.

A. Industry 4.0 Safety Validation Challenge

We discussed the potential for modular and contract-based reasoning to drive the structure of the overall safety case for Industry 4.0 architectures and meet the safety requirements. However, the fundamental problem does not lie in how the configurable architectures meet the safety requirements. Rather, the issue lies in the *generation* of these safety requirements in the first place. The ad hoc assemblage of Things and infrastructures for Industry 4.0 architectures will likely result in new hazards and/or risk ratings and as such new safety requirements. These *emerging* hazards are due to expected, yet unpredictable, reconfigurations or redeployments of the architecture in multiple contexts (i.e. we cannot assume that the world is stable, and variation only lies within our system). This will often mean that the hazard analysis, or at least a large part of it, will have to be manually repeated for each reconfiguration or deployment and should produce an updated set of safety requirements (i.e. each of these changes might be considered as a new factory). This can be seen as undermining the general-purpose and reusable nature of Industry 4.0 architectures, i.e. where rapid reconfiguration and deployment is seen as a unique selling point. In other words, modularity and contract-based reasoning largely deal with the *verification* issue whereas hazard analysis of the whole system addresses the *validation* problem. Safety validation, against the intended real-world usage, is the essence of safety assurance and how risk and harm are assessed, perceived and accepted.

B. Industry 4.0 Safety Confidence Challenge

In our example definition of assurance contracts for Things and infrastructures within Industry 4.0 architectures, we highlighted the need to specify necessary properties that have to be provided (e.g. measurement of air pressure values) to a particular level of integrity (e.g. accuracy of 0.001%) and confidence (e.g. 99%). For large socio-technical IoT systems such as Smart Factories, confidence will inevitably be measured using different qualitative [11] and quantitative [6] indicators. Propagating confidence from the different qualitative and quantitative measures associated with the various Things in an infrastructure is necessary to assess confidence in the safety of the overall configured system [10]. This has to be performed dynamically and on-demand to address the particular reconfigurable characteristics of Industry 4.0 architectures. This is a grand safety challenge for Industry 4.0 (and safety engineering generally). Current approaches to specifying confidence and associating it with assume-guarantee contract specification for individual components is relatively straightforward compared to the challenge of assessing, dynamically, confidence for the different reconfigurations.

C. Industry 4.0 Commercial Pressure Challenge

The financial appeal of commercially available Things and infrastructures, which *appear* to be dependable although they are not developed for safety-critical applications, should not be undermined. The business pressure is mounting on safety engineers to accept the use of, relatively *cheap*, consumer electronics and commercially available cloud-based services. Resistance from the safety community on the basis of difficulty or novelty could be counter-productive. This might result in alienating or excluding safety engineers when design decisions are made or more likely, and sometimes rightly so, appealing to reduction in overall risk despite increases in technological risks (e.g. a typical risk-benefit argument in clinical applications in which clinical benefits outweigh technological risks [13]).

D. Industry 4.0 Security-Informed Safety Challenge

There is now almost a consensus on the necessity to address cyber security in safety assurance [4]. This issue takes a greater significance for Industry 4.0 where remote connectivity and the use of commercially available infrastructures and Things expose the system to a wide range of cyber threats (particularly Distributed Denial of Service [9]). Security risks tend to be more dynamic than safety risks. As such, exploring the extent to which an Industry 4.0 architecture might have to reconfigure in the event of a security breach is a significant challenge, particularly in how it might compromise safety assurance (i.e. a typical trade-off between safety and security that has to be made more explicit in the safety assurance case).

In conclusion, in this paper, we explored a number of characteristics for the safety assurance of Industry 4.0 and focused on modularity as a key aspect of the overall assurance case for safety. We also highlighted some grand challenges that remain and will be a focus for our future work.

Terminology

XaaS – Anything (X) as a Service

Internet of Things (IoT) - a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

Thing devices – enable interaction with the physical environment via different sensors/actuators.

Cloud Computing – a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Edge Computing – a decentralized infrastructure in which parts of applications, management and data analytics are moved to the end devices such that computing is performed as close as possible to the data source.

Fog Computing - a decentralised infrastructure in which parts of applications, management and data analytics are moved into the network itself using a distributed computing model.

Fog/Edge Devices – local computational devices that offer advanced storage and processing power to the Things and rely on remote Cloud infrastructure for high-power computing and storage.

Acknowledgments This work is supported by the Swedish Foundation for Strategic Research (SSF) via the project Future factories in the Cloud (FiC).

References

[1] Goal Structuring Notation working group, November 2011.

[2] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[3] A. Bessani, R. Kapitza, D. Petcu, P. Romano, S. V. Gogouvitis, D. Kyriazis, and R. G. Cascella. A look to the old-world sky: EU- funded dependability cloud computing research. *Operating Systems Review*, 46(2):43–56, July 2012.

[4] R. Bloomfield, K. Netkachova, and R. Stroud. Security-informed safety: if its not secure, its not safe. In International Workshop on Software Engineering for Resilient Systems, pages 17–32. Springer, 2013.

[5] P. Conmy, M. Nicholson, and J. McDermid. Safety assurance contracts for integrated modular avionics. In *Proceedings of the 8th Australian workshop on Safety critical systems and software-Volume 33*, pages 69–78. Australian Computer Society, Inc., 2003.

[6] E. Denney, G. Pai, and I. Habli. Dynamic safety cases for through-life safety assurance. In *Proceedings of the 37th International Conference on Software Engineering-Volume 2*, pages 587–590. IEEE Press, 2015.

[7] B. Esmaeilian, S. Behdad, and B. Wang. The evolution and future of manufacturing: A review. *Journal of Manufacturing Systems*, 39:79 – 100, 2016.

[8] J. Fenn, R. Hawkins, P. Williams, and T. Kelly. Safety case composition using contracts-refinements based on feedback from an industrial case study. In *The Safety of Systems*, pages 133–146. Springer London, 2007.

[9] Guardian. DDoS attack that disrupted internet was largest of its kind in history, experts say. www.theguardian.com/technology/2016/oct/26/ ddos-attack-dyn-mirai-botnet.

[10] J. Guiochet, Q. A. Do Hoang, and M. Kaaniche. A model for safety case confidence assessment. In *International Conference on Computer Safety, Reliability, and Security*, pages 313–327. Springer, 2015.

[11] R. Hawkins, T. Kelly, J. Knight, and P. Graydon. A new approach to creating clear safety arguments. In *Advances in systems safety*, pages 3–23. Springer, 2011.

[12] W. He and L. Xu. A state-of-the-art survey of cloud manufacturing. *Int. J. Comput. Integr. Manuf.*, 28(3):239–250, Mar. 2015.

[13] ISO. *ISO 14971: medical devices-application of risk management to medical devices*. ISO, 2012.

[14] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster. *Recommendations for Implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry*. Forschungsunion, 2013.

[15] T. P. Kelly. *Arguing safety: a systematic approach to managing safety cases*. University of York, 1999.

[16] T. P. Kelly. Concepts and principles of compositional safety case construction. *Contract Research Report for QinetiQ COMSA/2001/1/1*, 34, 2001.

[17] P. Mell, T. Grance, et al. The nist definition of cloud computing. 2011.

[18] PILZ. Industrie 4.0 – safe and smart (white paper), June 2016.

[19] B. P. Rimal, E. Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 44–51, Aug 2009.

[20] J. Rushby. Modular certification. Technical report, Sept. 2001.

[21] J. Rushby. The interpretation and evaluation of assurance cases. Technical Report SRI-CSL-15-01, Computer Science Laboratory, SRI International, Menlo Park, CA, July 2015. Available at http://www.csl. sri.com/users/rushby/papers/sri-csl-15-1-assurance-cases.pdf.

[22] J. Rushby. Trustworthy self-integrating systems. In N. Bjørner, S. Prasad, and L. Parida, editors, *12th International Conference on Distributed Computing and Internet Technology, ICDCIT 2016*, volume 9581 of *Lecture Notes in Computer Science*, pages 19–29, Bhubaneswar, India, Jan. 2016. Springer-Verlag.

[23] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone. Taming dr. frankenstein: Contract-based design for cyber-physical systems. *European journal of control*, 18(3):217–238, 2012.

[24] D. Schneider and M. Trapp. Conditional safety certification of open adaptive systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 8(2):8, 2013.

[25] Telecommunication standardization sector of ITU. *Overview of the Internet of things*, Y.2060 edition, 6 2012.

[26] D. Wu, M. J. Greer, D. W. Rosen, and D. Schaefer. Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems*, 32(4):564 – 579, 2013.