Integrating Safety Assessment into the Design of Healthcare Service-Oriented Architectures

Ibrahim Habli[1], Abdulaziz Al-Humam[1], Tim Kelly[1], and Leila Fahel[2]

1 **University of York, York, UK**
{Ibrahim.Habli,aaah501,Tim.Kelly}@york.ac.uk
2 **Calderdale and Huddersfield National Heath Service Foundation Trust, Halifax, UK**
Lfahel@nhs.net

Abstract

Most healthcare organisations are service-oriented, fundamentally centred on critical services provided by medical and nursing staff. Increasingly, these human-centric services rely on software-intensive systems, i.e. medical devices and health informatics, for improving different aspects of healthcare, e.g. enhancing efficiency through automation and patient safety through smart alarm systems. However, many healthcare services are categorised as high risk and as such it is vital to analyse the ways in which the software-based systems can contribute to unintentional harm and potentially compromise patient safety. This paper proposes an approach to modelling and analysing Service-Oriented Architectures (SOAs) used in healthcare, with emphasis on identifying and classifying potential hazardous behaviour. The paper also considers how the safety case for these SOAs can be developed in a modular manner. The approach is illustrated through a case study based on three services: ambulance, electronic health records and childbirth services.

1998 ACM Subject Classification K.4.1 [Public Policy Issues]: Computer-related health issues

Keywords and phrases Healthcare, Safety, Assurance, Service-Oriented Architecture

Digital Object Identifier 10.4230/OASIcs.MCPS.2014.113

1 Introduction

Healthcare organisations are structured based on different, yet interdependent, services [1], e.g. emergency and urgent care services, cancer services and ambulance services. Many of these services are supported by software-intensive systems. These systems include medical devices [23] (e.g. infusion pumps) and networked health IT systems (e.g. distributed Electronic Health Record (EHR) systems). There is also an increased interest in improving the integration between the different healthcare services through enhancing software and data interoperability and standardising the interfaces between the health IT infrastructures and medical devices [2, 3, 4].

Despite their significant benefits, software-based services and systems can pose risks to patient safety [5, 6]. For example, between 2005 and 2009, the US Food and Drug Administration (FDA) received over 56,000 reports of issues related to the use of infusion pumps [7]. Many of the safety issues were traced to software defects. In the UK, the Medicines and Healthcare Products Regulatory Agency (MHRA) reported a continuous increase in medical device adverse incidents, totalling 9099 reports in 2009 [8]. The British Medical Journal (BMJ) also reported a significant increase in medical device recalls and warnings [9]. Given the criticality of certain software systems, e.g. EHR, assessing the extent to which

Medical Cyber Physical Systems – Medical Device Interoperability, Safety, and Security Assurance (MCPS'14).
Editors: Volker Turau, Marta Kwiatkowska, Rahul Mangharam, and Christoph Weyer; pp. 113–123

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the software behaviour contributes to safety hazards in healthcare services should be an integral part of the clinical risk assessment process and the overall clinical safety case [10, 11]. These safety hazards arise in clinical environments that are centred on the interactions between many different human, procedural and technological elements. Understanding and controlling the complex links between the software behaviour and the emergence of the clinical hazards (i.e. potential to cause preventable/unintentional harm) is a significant challenge. Addressing this challenge at the clinical level requires close collaboration between different stakeholders, primarily clinical experts, health scientists, safety analysts and system and software engineers. At the level of software systems, software engineers need to analyse failures within, and between, the software-intensive healthcare systems (e.g. incorrect dosage information provided automatically to an infusion pump by the EHR system [2]). Jointly with clinical experts, these engineers should also analyse how these failures can become hazardous behaviours once situated within a clinical environment (e.g. inaccurate blood pressure data due to physical factors such as the height of an IV bag [12]). Unfortunately, inadequate interaction between clinical experts and software engineers remains a major hurdle for achieving effective risk assessment of software-intensive healthcare services [13].

Importantly, assurance has to be provided that the risk of the software hazardous behaviours, identified in the clinical risk assessment process, has been adequately mitigated. Increasingly, this is being communicated in the form of a safety or assurance case [7, 10, 11]. Safety cases provide a reasoned and structured argument of how the available evidence, generated from testing and analysis, supports overall claims made about system safety. For a complex clinical environment, the overall safety case is not monolithic but compositional [14], comprising different safety arguments and evidence for the various healthcare services. These services, including systems and processes developed by different organisations, are clearly interdependent and so are their corresponding safety arguments [21].

This paper focuses on the safety assessment of services within healthcare Service-Oriented Architectures (SOAs). It presents a preliminary approach to modelling and analysing SOAs used in healthcare, with emphasis on identifying and classifying potential hazardous software behaviour. The approach is based on two existing modelling techniques for specifying individual services, and the processes connecting them, namely the Service oriented architecture Modelling Language (SoaML) [15] and the Business Process Modelling Notation (BPMN) [16]. This approach also builds on adapting two existing safety analysis techniques, namely the Functional Hazard Assessment (FHA) [17] and Software Hazard Analysis and Resolution in Design (SHARD) [18]. Further, the paper explores how the safety case for the SOA can be developed in a modular manner using the Goal Structuring Notation (GSN) [19]. The approach is illustrated through examples from an exploratory case study based on three services: ambulance, electronic health records and childbirth services.

The rest of the paper is organised as follows. An overview of the proposed SOA safety assessment approach is presented in Section 2, followed by a more detailed description of the safety analysis techniques in Sections 3 and 4. The safety analysis is illustrated in Section 5 using an exploratory case study. The nature and potential structure of a modular safety case for SOA are discussed in Section 6, followed by conclusions in Section 7.

2 SOA Safety Assessment

A service has been defined as "*a value delivered to another through a well-defined interface and available to a community*" [15]. The value that a service delivers, and the safety risks associated with it, can only be realised and understood in the sociotechnical setting of the

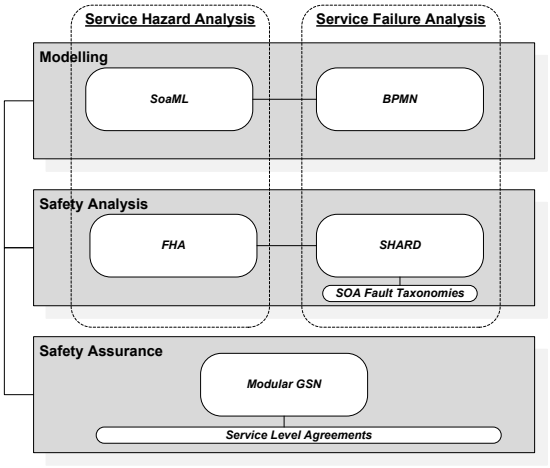

Figure 1 SOA Safety Assessment.

service (e.g. interactions between different types of services and processes). One approach to designing and representing this setting, especially for software-based services, is through an SOA. An SOA "*provides a paradigm for defining how people, organizations, and systems provide and use services to achieve results*" [15].

For healthcare services, the SOA paradigm can assist in the assessment of patient safety, in which accidents predominantly occur as a result of the interaction of different human, system and organisational behaviours. An overview of the SOA safety assessment approach proposed in this paper is depicted in Figure 1, and is briefly introduced in the rest of this section. A more detailed description of the safety analyses is provided in the next two sections.

Healthcare services, including interfaces and contracts between these services, are modelled in this approach using SoaML. The SoaML models depict a modular description of the healthcare SOA in which related contracts, interfaces and operations are encapsulated in, and provided by, self-contained services. In order to identify the hazards associated with services, we propose a variant of FHA, called Service Hazard Analysis (SHA), based on analysing three potential service deviations [17]: (1) *service not provided when required*, (2) *service provided when not required* and (3) *incorrect service*. The primary output of SHA is a set of safety requirements defined at the service level.

Next, in order to analyse the causes of the service hazards, identified using SHA, the detailed tasks implementing the SOA services and processes are modelled in BPMN, focusing particularly on the flow of information between interacting tasks. The SHARD safety analysis technique [18], which is a variant of the hazard and operability study (HAZOP) technique [20], is adapted to analyse the flow of information between the tasks represented within the SOA processes. We refer to this as Service Failure Analysis (SFA). The analysis and resulting failure modes are driven by the application of a set of guidewords: *omission, commission, early, late* and *incorrect value* [18]. Each of the failure modes is then associated with specific service faults linked to existing SOA fault taxonomies (i.e. to make the analysis SOA-specific [25]). The output of this analysis is a set of derived service safety requirements.

Finally, the above analyses are used as a core part of the safety evidence base to inform the structure of the overall safety argument for the SOA safety case. Given the modular nature of SOA, the safety argument is structured based on different, yet interrelated, argument modules [14]. Most of these argument modules correspond to the safety justification of a

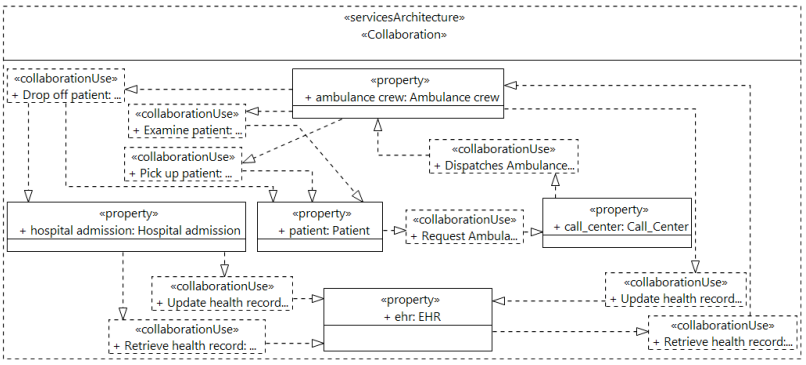

Figure 2 ServicesArchitecture Model.

specific individual service. Typically, within SOA, services interact based on pre-defined Service-Level Agreements (SLAs) [22]. Similarly, the relationship between different argument modules can be specified using argument contracts [14]. The mapping between SLAs and the argument contracts drives the overall structure of the SOA safety argument and offers traceability between the design, represented in the SOA models, and the safety assurance, represented in the modular safety argument.

3 Service Hazard Analysis (SHA)

Safety analysis processes are centred on identifying, analysing and managing hazards. For healthcare SOA, identifying the hazards associated with the services is essential for defining the service safety requirements, which should influence the architectural design. In this paper, hazard identification and classification is carried out based on SHA, using a high-level representation of the SOA in SoaML. SHA consists of four steps:

1. **Identify a service:** high-level services are captured in SoaML *ServicesArchitecture* models. A *ServicesArchitecture* represents how *Participants* collaborate, by producing and consuming *Services* to achieve goals. Figure 2 shows an example *ServicesArchitecture* model that captures *Participants* (e.g. *ambulance crew*, *patient* and *hospital-admission*) and *Services* (e.g. *Request ambulance*, *Examine patient* and *Update health record*).
2. **Identify the service failure modes:** the modes or types of failures that are considered in this step are as follows: (1) *service not provided when required*, (2) *service provided when not required* and (3) *incorrect provision of service*. These are intended for use as prompts for identifying the different ways in which the service can fail. The use of these types requires the safety analyst to interpret the meaning and relevance of specific failures in the context of the service in hand (e.g. *incorrect drug dosage* has to be interpreted in the context of specific classes of drugs for specific conditions or combinations of conditions).
3. **Determine the safety effects and severity of each service failure mode:** the adverse consequences of the failure mode should be determined, taking into consideration different factors, e.g. the condition of the patient and other services and systems (not just the software systems). A safety classification should also be defined (e.g. *Catastrophic, Major, Considerable* or *Significant*), typically based on Hazard/Risk Matrices (HRM) defined in standards or by the healthcare organisation [10, 11]. In terms of determining the effects of service failures, a useful feature of SoaML *ServicesArchitecture* models is that they include a representation of service interaction and the *Participants* that use these services (i.e. making it easier for the analyst to trace the effects of a failure mode). For example,

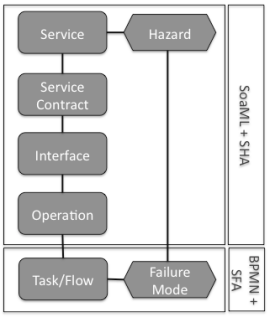

Figure 3 Link between SoaML and BPMN Models.

administrating certain drugs or treatments (e.g. radiotherapy) might have potential hazardous effects on both the patients and the caregivers (both represented as SoaML *Participants*).

4. **Provide recommendations or service safety requirements:** based on the identified failure modes and their classification, recommendations, preferably in the form of service safety requirements, should be generated, where the rigor/integrity with which the requirements need be met should be proportionate to the severity of the failures (i.e. higher degrees of severity requires more stringent integrity requirements and more rigorous processes) [24, 27]. SHA is implementation-independent and should be performed at the early specification and design stages of the SOA.

4 Service Failure Analysis (SFA)

Service hazards identified using SHA typically emerge from a combination of factors, whether human, organisational or technological. Understanding and analysing the causes of these hazards is an important step towards eliminating or reducing the risk of these hazards. In this section we introduce SFA to examine the detailed implementation of the services, and identify how failures, specifically interaction/information flow failures, can contribute to service hazards. SFA consists of five steps:

1. **Identify a flow between two tasks:** SFA, as proposed in this paper, is based on a behavioural model of the SOA represented in BPMN [16]. BPMN provides the ability to communicate and represent internal procedures, based on *Processes*, using a graphical and structured notation. Typically, these *Processes* represent workflows of connected *Tasks* (i.e. atomic *Activities*), grouped into *Swimlanes* (i.e. a container for organising *Activities*). Figure 4 shows an example BPMN process model, comprising connected *Tasks* that are organised into five different *Swimlanes* (e.g. *ambulance crew*, *patient* and *hospital-admission*). The SoaML model used for SHA (Section 3) and the BPMN model used for SFA, in this section, are linked are follows (Figure 3): Services in SoaML interact through defined *Contracts*. These *Contracts* have explicit *Interfaces* that offer a number of *Operations*. Each of these *Operations* is then linked to a *Task* in BPMN, enabling traceability between the different models at different abstraction levels. Further, each *Participant* in SoaML is represented as a *Swimlane* in BPMN. This step in SFA involves the selection of a link between two BPMN Tasks that captures a flow in terms of inputs, outputs, data, sequence and timing.
2. **Identify flow failure modes:** similar to Step 2 in SHA, this step uses a number of guidewords, this time based on SHARD, to determine the ways in which the selected

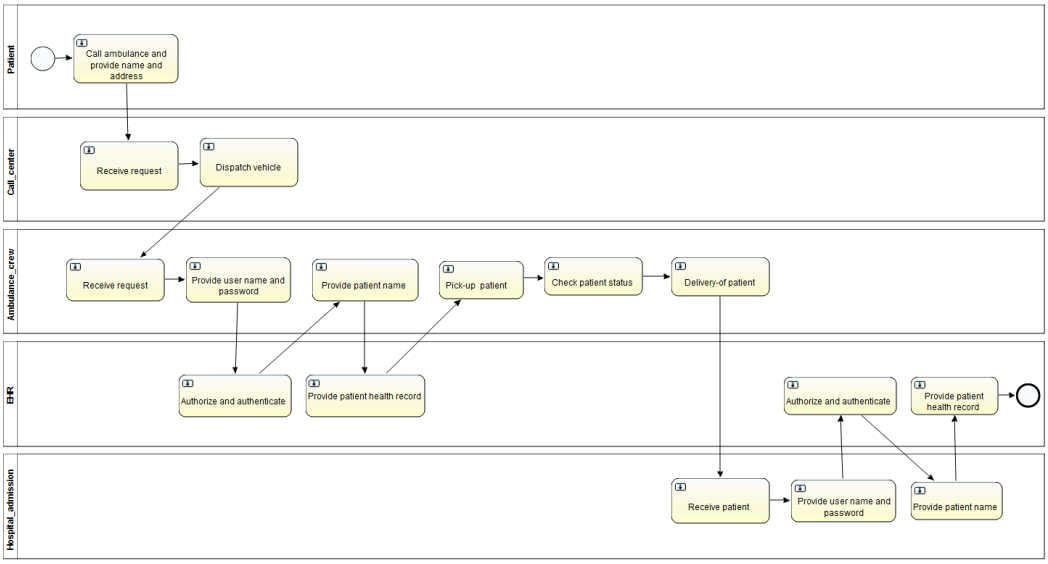

Figure 4 BPMN Model.

flow can deviate from its intended usage. Five guidewords are used here: *omission, commission, early, late* and *value* [18]. The failures derived from the use of each of these guidewords should be interpreted in the context of the SOA design.

3. **Determine the potential causes of each flow failure mode:** causes can be a combination of technical, human and organisational events or conditions. For technical causes in particular, we use the SOA fault taxonomy developed by Bruning et al [25]. This fault taxonomy is well structured and categorises faults based on the SOA lifecycle phase in which they can emerge: (1) publishing, (2) discovery, (3) composition, (4) binding and (5) execution faults. The advantage of using these fault types is that they are SOA-specific. However, again, these fault types should be used as prompts or hints for safety analysts rather than as an exhaustive list of all possible SOA faults.
4. **Determine the potential effects of each flow failure mode:** the potential effects should be recorded and should be examined in terms of the contribution that they can make to the hazardous service failure modes, identified in SHA, or possibly the contribution that they might make to new hazardous behaviours (i.e. missed during SHA).
5. **Provide detailed safety requirements or design recommendations:** where a failure mode contributes to one or more hazards, one or more safety requirements should be defined to address the failure mode. Further, some design recommendations could be made for addressing the failure mode, e.g. based on existing safety tactics in the software architecture literature or SOA-specific dependability tactics (e.g. in [26], which are centred on the use of service redundancy, diversity, graceful degradation, monitoring and containment).

5 Exploratory Case Study

In this section, we illustrate the use of SHA and SFA using extracts from an exploratory case study, which is based on three healthcare services: ambulance, EHR and childbirth services.

A subset of these services is represented in the SoaML *ServicesArchitecture* model shown in Figure 2. It covers the services used (i.e. produced and consumed) from the point a phone

Table 1 SHA Results.

Service	Failure Mode	Effects	Class ([7])	Recommendation
Dispatch Ambulance *(Context: birth before attendance and the patient is actively bleeding following birth)*	Ambulance not dispatched	Patient death	Major	Active monitoring and cross-checking between requested and dispatched ambulances.
	Ambulance dispatched when not required	N/A	N/A	No direct safety effects but waste of critical resources.
	Ambulance dispatched later than intended	Severe morbidity (hypovolaemia, renal failure, cardiac arrest, disseminated intravascular coagulopathy…)	Major	Active monitoring of timing targets and strategies for recovery from timing-related failures.
	Ambulance dispatched to the wrong address	Severe morbidity (hypovolaemia, renal failure, cardiac arrest, disseminated intravascular coagulopathy…)	Major	Early address cross-checking and confirmation between requested and dispatched ambulances
…	…	…	…	…

call is made to request an ambulance (for a pregnant woman) to the point at which the patient is admitted to a hospital (labour ward). Another set of *ServicesArchitecture* models was created to capture subsequent stages e.g. fetal monitoring, first and second stages of labour, caesarean section and postnatal care (not discussed further in this paper).

Table 1 shows an extract from the SHA results when applied to the *Dispatch Ambulance* service. The analysis establishes the potential safety criticality of the *Dispatch Ambulance* service, based on the severity of the worst credible effects of the identified failure modes, leading to stringent safety requirements allocated to the *Dispatch Ambulance* service.

As can be observed from the results, the analysis is collaborative in nature, demanding inputs from both engineers (e.g. determining how the service can fail) and clinicians (e.g. assessing effects on patients). Figure 4 presents a more detailed model of the SOA using BPMN, with more emphasis of the SOA process. BPMN Tasks in this process are mapped into the *Operations* in the *Interfaces* provided by each SoaML *Service* while BPMN *Swimlanes* are mapped onto the SoaML *Participants*.

Considering each flow between the *Tasks*, SFA was applied to the BPMN model. A sample outcome is shown in Table 2, considering the last flow in the BPMN model, from *Provide patient health record* to the End object (i.e. admission to labour ward). The analysis shows how the lack of patient information from the EHR, and more seriously incorrect information, can potentially lead to adverse patient complications.

6 SOA Safety Cases

Establishing and justifying an acceptable level of confidence in the safety of software-based healthcare services will often require different safety arguments and evidence generated by different service owners or providers. One approach to representing this compositionality of the overall safety case for service-based systems is through modular GSN [19]. Modular GSN supports the definition of the safety case based on the composition of different, but

Table 2 SFA Results.

Flow	Failure Mode	Causes	Effects	Recommendation
Link between 'Provide patient health record' to the End object	No record available (Omission)	Authentication failed or request timed out (server crashed)	Incomplete history and background (mild) Anaphylaxis-unknown allergy status (severe) …	Use of redundancy in data sources for health records
	Incorrect record retrieved (Value)	Incorrect input or conversion fault	Anaphylaxis-unknown allergy status (severe)	Data entry cross-checking, online monitoring and fault containment.
	Early	N/A	N/A	N/A
…	…	…	…	…

interrelated, argument modules. When argument modules are composed a record of the agreement and consistency can be recorded using a safety case contract [14]. This contract "*contains definition of the relationships between two modules, defining how a claim in one supports the argument in the other*" [19].

Modularity in the definition of services lends itself to the concept of modular safety cases. Service owners or producers often rely on other services when guaranteeing and providing their own services. Similarly, claims in certain argument modules can only be said to be substantiated (the guarantee clause) if claims or evidence are available in other argument modules that offer sufficient support (the rely clause). Figure 5 shows a preliminary GSN modular structure for the safety case for the healthcare SOA considered in the case study in the previous section. The safety case has three categories of argument modules:

- *Top-Level Argument Module* includes a hazard-directed argument, which covers the main safety claims concerning the identified service hazards, including interaction hazards;
- *Ambulance Service Argument Module, EHR Service Argument Module, Hospital Admission Service Argument Module* and *Service Interactions Argument Module* include hazard mitigation arguments for the hazards posed by the different services and their interactions; and
- *Ambulance Crew and Call Centre Argument Module, EHR System Argument Module* and *Hospital Admission Argument Module* include detailed arguments concerning the systems and organisations responsible for implementing the healthcare services.

However, the safety case structure in Figure 5 and the case study description in Section 5 do not take into account that in larger regions (which include sub-regions), the same types of healthcare services can be offered by a variety of different ambulance service providers and hospitals. In such situations, the high-level argument structure in Figure 5, comprising the top-level hazard-directed argument module and hazard mitigation argument modules potentially need not change. Where the implementation of the provision of services changes, the bottom tier safety argument structure will need to change. For example, if a healthcare region was divided into two sub-regions, say east and west, with different hospitals, ambulance crews and call centres, then the safety arguments concerning the systems and organisations implementing the healthcare services would be different, taking into account the specific design and operational issues concerning these systems and organisations.

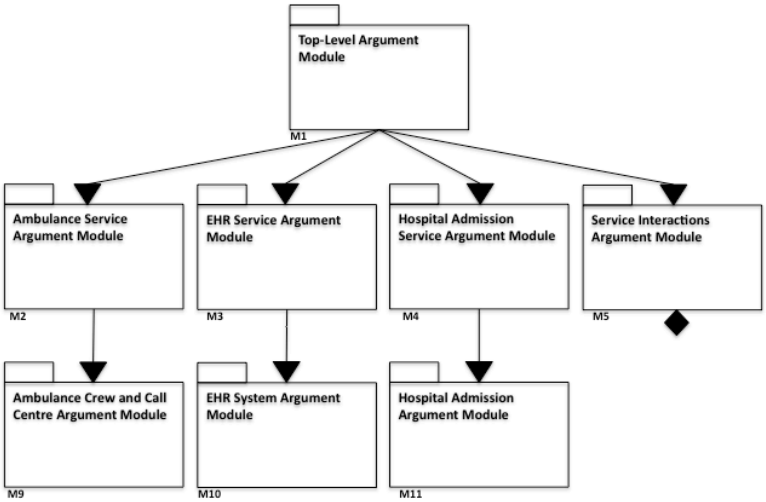

Figure 5 SOA Safety Argument.

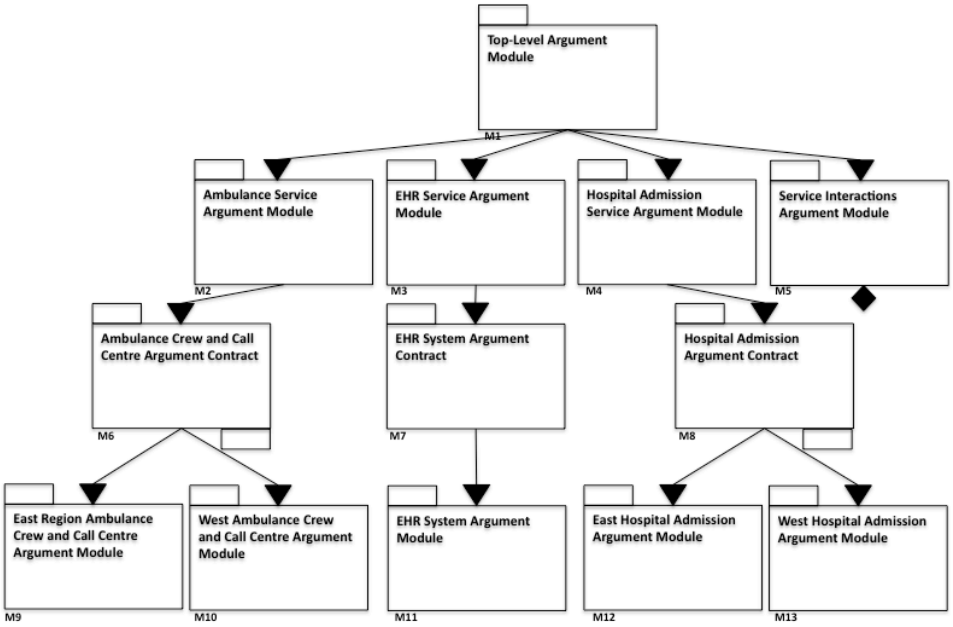

Figure 6 SOA Safety Argument (including Contracts).

Figure 6 shows a modified representation of the safety case structure in Figure 5, taking into consideration the need for two different safety arguments (east and west) for each of these argument modules: *Ambulance Crew and Call Centre Argument Module* and *Hospital Admission Argument Module*. However, it can be noticed that the *Ambulance Service Argument Module* and *Hospital Admission Service Argument Module* are now supported by the *Ambulance Crew and Call Centre Argument Module* and *Hospital Admission Argument Module* (east and west) via two contract modules. This loose coupling between these argument modules can help in minimising the impact of change to the safety case when different systems and organisations are used to provide the services (e.g. a third sub-region is introduced).

7 Conclusions

This paper has presented a preliminary approach to integrating safety assessment into the design of healthcare SOA, covering three aspects: modelling using SoaML and BPMN, safety analysis using SHA and SFA (adapting FHA and SHARD) and safety assurance using modular GSN. An exploratory case study was also discussed, based on three services: ambulance, electronic health records and childbirth services. We are currently developing a tool-support platform for the above safety assessment approach in order to improve traceability between the design, safety analysis and safety assurance models and provide automated means for supporting the safety analysis process. We are also developing a set of modular safety argument patterns that analysts can use as a basis for structuring SOA safety arguments.

Finally, examining the safety impact of interactions between services and service providers remains a significant challenge, especially for healthcare services that span both primary and secondary care and cover more than one medical condition (e.g. care and treatment of diabetes in pregnancy which involves GPs, obstetricians, midwives, endocrinologists, and diabetes-specialist nurses). Our future work will examine means for identifying, modelling and analysing these interactions by integrating search-based technologies (e.g. simulation and model-checking) into the above SOA safety assessment approach.

References

1 Keen J. *What is a care pathway?* 4th International Workshop on Software Engineering in Health Care, Zurich, Switzerland, June 2012

2 Sokolsky, O., Lee, I., and Heimdahl, M. *Challenges in the regulatory approval of medical cyber-physical systems*. International Conference on Embedded Software, Taipei, Taiwan, October 2011

3 Arney, D., Goldman J. M., Whitehead, S. F., and Lee, I. *Synchronizing an X-ray and anesthesia machine ventilator: a medical device interoperability case study*. International Conference on Biomedical Electronics and Devices, Porto, Portugal, January, 2009

4 IEC. *IEC 80001-1:2010, Application of Risk Management for IT-Networks Incorporating Medical Devices – Part 1: Roles, Responsibilities and Activities*. IEC, October, 2010

5 Rakitin, R. *Coping with defective software in medical devices*. IEEE Computer. 39, 4, April 2006

6 Koppel, R., Metlay, J. P., Cohen, A., Abaluck. B., Localio, A. R., Kimmel, S. E., and Strom, B. L. *Role of computerized physician order entry systems in facilitating medication errors*. The Journal of Urology, March, 2005

7 FDA. *Total Product Life Cycle: Infusion Pump Premarket Notification 510(k) Submissions*. April 2010

8 MHRA. *Adverse Incident Reports 2009*. Device Bulletin DB2010 (03), 2009

9 Heneghan, C., Thompson, M., and Billingsley, M. *Medical device recalls in the UK and the device regulation process: retrospective review of safety notices and alerts*. BMJ, May 2011

10 Health and Social Care Information Centre. *Clinical Risk Management: its Application in the Manufacture of Health IT Systems*. ISB 0129, 2013

11 Health and Social Care Information Centre. *Clinical Risk Management: its Application in the Deployment and Use of Health IT Systems*. ISB 0160, 2013

12 Hofmann, R. *Modeling Medical Devices for Plug-and-Play Interoperability*. MS Thesis, MIT, 2007

13 Sujan, M., Koornneef, F., Chozos, N., Pozzi, S., and Kelly, T. *Safety cases for medical devices and health IT: involving healthcare organisations in the assurance of safety*. Health Informatics Journal, 18, 4, September 2013

14 Fenn, J., Hawkins, R., Kelly, T., and Williams, P. *Safety case composition using contracts: refinements based on feedback from an industrial case study*. Safety-Critical Systems Symposium, Bristol, UK, February 2007

15 OMG. *Service oriented architecture Modeling Language (SoaML) Specification*. Version 1.0.1, May 2012

16 OMG. *Business Process Model And Notation (BPMN)*. Version 2.0, 2011

17 SAE. *ARP4761. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. December 1996

18 Pumfrey, D. *The Principled Design of Computer System Safety Analyses*. PhD Thesis, The University of York, September 1999

19 GSN Standard Committee. *Goal Structuring Notation (GSN)*, [On-line]. `http::www.goalstructuringnotation.info`

20 Kletz T. *Hazop and Hazan*. 4th ed., Taylor and Francis, 2006

21 Brown, A., Fenn, J., and Menon. *Issues and considerations for a modular safety certification approach in a service oriented architecture*. IET International System Safety Conference, 2010

22 Keller, A., and Ludwig, H. *The WSLA framework: specifying and monitoring service level agreements for web services*. Journal of Network and Systems Management, 11, 1, March 2003

23 ISO. *ISO 14971:2012: Medical devices. Application of Risk Management to Medical Devices*. July 2012

24 Habli, I., Hawkins, and R., Kelly. *Software safety: relating software assurance and software integrity*. International Journal of Critical Computer-Based Systems (IJCCBS). 1, 4, November 2010

25 Bruning, S., Weissleder, S., and Malek. *A fault taxonomy for service-oriented architecture*. High Assurance Systems Engineering Symposium, Dallas, US, 2007.

26 Buckley, I., Fernandez, E.B., Anisetti, M., Ardagna, C., Sadjadi, M., and Damiani, E. *Towards pattern-based reliability certification of services*. On the Move to Meaningful Internet Systems, Hersonissos, Greece, 2011.

27 Hawkins, R., Habli, I., and Kelly, T. *The Principles of Software Safety Assurance*. 31st International System Safety Conference, Boston, USA, August 2013