Adapting a System-Theoretic Hazard Analysis Method for Interoperability of Information Systems in Health Care

by

Oscar Aleixo Costa Rocha
B.Sc., Centro Universitário de Belo Horizonte, Brazil, 2006

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In the Department of Computer Science

Adapting a System-Theoretic Hazard Analysis Method for Interoperability of Information Systems in Health Care

by

Oscar Aleixo Costa Rocha
B.Sc., Centro Universitário de Belo Horizonte, Brazil, 2006

Supervisory Committee

Dr. Jens Weber, Supervisor
(Department of Computer Science, University of Victoria, BC)

Dr. Morgan Price, Co-supervisor
(Department of Computer Science, University of Victoria, BC)

ABSTRACT

The adoption of Health Information Systems (HIS) by primary care clinics and practitioners has become a standard in the healthcare industry. This increase in HIS utilization enables the informatization and automation of many paper-based clinical workflows, such as clinical referrals, through systems interoperability. The healthcare industry defines several interoperability standards and mechanisms to support the exchange of data among HIS. For example, the health authorities, Interior Health and Northern Health, created the CDX system to provide interoperability for HIS across British Columbia using SOAP Web Services and HL7 Clinical Document Architecture (CDA) interoperability standards. The CDX interoperability allows HIS such as Electronic Medical Record (EMR) systems to exchange information with other HIS, such as patients clinical records, clinical notes and laboratory testing results. In addition, to ensure the EMR systems adhere to the CDX specification, these health authorities conduct conformance testing with the EMR vendors to certify the EMR systems. However, conformance testing can only cover a subset of the systems' specifications and a few use cases. Therefore, systems properties that are not closely associated with the systems (i.e. emergent properties) are hard, or even impractical, to assure using only conformance testing. System safety is one of these properties that are particularly significant for EMR systems because it deals with patient safety. A well-known approach for improving systems safety is through hazard analysis. For scenarios where the human factor is an essential part of the system, such as EMR systems, the System-Theoretic Process Analysis (STPA) is more appropriate than traditional hazard analysis techniques. In this work, we perform a hazard analysis using STPA on the CDX conformance profile in order to evaluate and improve the safety of the CDX system interoperability. In addition, we utilize and customize a tool named FASTEN to support and facilitate the analysis. To conclude, our analysis identified a number of new safety-related constraints and improved a few other already specified constraints.

Table of Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My parents, Oscar N. Costa and Esmeralda B. Rocha for providing many opportunities and support, which allowed me to achieve my objectives.

Jens Weber for his invaluable supervision, support, encouragement, and patience.

Morgan Price for his insightful comments and suggestions.

Daniel Ratiu, and the MPS Slack server members for the help on JetBrains MPS and FASTEN.

Chapter 1

Introduction

The adoption of Health Information Systems (HIS), such as Electronic Medical Record (EMR) systems by primary care clinics and practitioners, has become a standard in the healthcare industry of many countries [41]. This increase in demand leads to the emergence of new suppliers and new products, which leads to innovation and technological advances. An example of innovation in healthcare is the advent of electronic referral (eReferral), which replaces paper-based clinical referral. This work focuses on the *Clinical Data eXchange* (CDX) system, which is a clinical document distribution service capable of provide the eReferral and other clinical document workflows [27].

eReferral and other forms of information exchanging in healthcare are only possible due to system interoperability, which is the capability of information systems to exchange and use the information. However, successfully implementing interoperability is not an easy task for many reasons [6]. For example, the heterogeneity of the information systems on handling data is a common problem that needs to be solved to achieve interoperability. A solution for this type of obstacle is adopting interoperability standards for information exchange. Interoperability standards define a *lingua franca* that allows different systems to communicate seamlessly and thus achieve interoperability.

The healthcare industry employs various interoperability standards for health information exchange, allowing systems to share information regardless of the system or supplier [20]. In addition, those standards are usually created by workgroups composed of various players of the industry and governments to ensure the agreement within the industry [52, 6]. These interoperability standards provide safe means to exchange information among systems. However, only implementing standards does not ensure that the interoperability of the systems is working in a safe manner be-

cause safety-related issues could arise at different levels of interoperability. Especially at the higher levels that are not fully assessed by the standards.

Some sector deals with the exchange of sensitive information, such as patient clinical records in the healthcare industry. HIS that handle this kind of information are considered safety-critical systems because any problem, including interoperability issues, could jeopardize the patients' health [64]. In these cases, conformity assessment or certifications are often utilized to ensure the safety of the systems [65]. Moreover, these certifications are comprised of conformance tests, or conformance profiles, which are formulated straight from the system specifications. However, testing the entire set of requirements is not always possible or practical. Therefore, conformance profiles are usually restrained to a sub-set of the specification [42]. This scope restriction could compromise the system safety because safety-related requirements were unintentionally left out from the conformance profiles. In addition, if any safety-related constraint was not foreseen in the original system requirements, the conformance profiles will also lack these constraints, hence jeopardizing the safety of the system.

Hazard analysis techniques can be utilized to draft safety-related requirements, or even design concepts, in the early phases of system development, which is recommended in order to build safe systems [34]. Nevertheless, information systems are in constant evolution, incorporating new features and services throughout the life of the systems. These changes could lead to safe-related issues and other problems on information systems if safety concerns are not considered. In this work, we investigate the application of a hazard analysis technique into conformance profiles in order to ensure the safety of interoperability services for EMR systems.

1.1 Terminology

Before advancing any further, it is worth reviewing the definition of some terminology utilized in this thesis since these terms can have distinct meanings for different domains.

Interoperability is defined by the IEEE [25] as *"the ability of two or more systems or components to exchange information and to use the information that has been exchanged."* This definition summarizes well what other domains advocate on interoperability of information systems. For instance, the Healthcare Information and Management Systems Society (HIMSS) [20] establishes a similar definition of interoperability, with the addition that the exchanged data is utilized in a coordinated

manner among different organizational levels and promptly ported in order to improve individual and population health.

Socio-technical system is a complex system, i.e. a system formed by multiple systems, in which people and technology interactions are so intense that it is not easy to design or analyze the organizational and technical systems independently [6].

System Safety is a form of preemptive engineering in which potential hazards are identified and evaluated to be addressed before they occur [35].

Loss involves anything that has value for the system's stakeholders [34]—I.e. Something that the stakeholders are unwilling to lose. It can be something abstract, like the system's or company's reputation, or something more concrete, such as property damage or human life.

Hazard can be defined as *"a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss"* [34]. Similar to the losses, hazard identification must be limited by the purpose of the analysis and realistic within the system's boundaries.

System-level hazard is a hazard that is delimited by boundaries of the system under investigation. Nevertheless, system-level hazards and losses should be clearly defined according to the analysis's objectives. These artifacts are essential to explain the purpose of the hazard analysis, as we will see in detail in Chapter 3.

1.2 Hazard Analysis Methods

Hazard analysis methods have been utilized since the 1940s and 1950s on military and aeronautic systems, and then on other mechanical and electronic systems [58]. Many of those methods are based on linear event models, as are the types of systems they were initially intended to address. Nonetheless, even with the advent and dissemination of complex socio-technical systems, those same methods continue in place as the most utilized.

Traditional hazard analysis methods like *Fault Tree Analysis* (FTA), *Event Tree Analysis* (ETA) are based on decision trees and focus on quantitative analysis through statistical methods [14]. Another event-based method is *Failure Mode and Effects Analysis* (FMEA) which was created for reliability analysis and adapted for hazard analysis [37]. FMEA uses a linear chain of events technique to analyze each step of the process (or system) for potential failures.

Hazard and Operability Analysis (HAZOP) is a system-based method that focuses

on investigating deviations in systems and processes design intent [14]. The method approach is to review the design using guide words and expertise on the system or process analyzed. Another system-based method is *Functional Resonance Accident Model* (FRAM), which focuses on how conditions that lead to accidents may occur from the variability in the system functions [23].

System Theoretic Process Analysis (STPA) is a method based on system theory that focuses on the dynamic behaviours of the system or the interaction between the system's components [34]. STPA uses a functional system model to analyze the system as a whole and focuses on the system's emergent properties. With this approach, STPA also incorporates the socio-technical aspects of the system in the analysis, which is hard to address on traditional event-based methodologies [34, 57].

1.3 Problem Definition

CDX is a clinical document distribution service that intends on facilitating the exchange of clinical documents among EMR systems across British Columbia, Canada [27]. Although all EMR systems are in the same domain (healthcare) and deal with the same type of data (patient clinical information), each system manipulates patient information in its own way. Thus, in order to make those systems interoperable, CDX works as a centralized communication hub and employs a bi-directional approach for exchanging data using the HL7 Version 3 Clinical Document Architecture (CDA) standard.

The EMR vendors that utilize the CDX system implement the CDA standard in their EMR systems. Then the EMR system is submitted to a conformance certification to demonstrate it is apt to send and receive documents through the CDX system. The certification is guided by a conformance profile that defines a set of criteria to satisfy technical, business and clinical requirements. However, it is unlikely that conformance tests cover all possible scenarios and outcomes [42]. Moreover, thinking from a safety perspective, any issue that happens throughout the information exchange using the CDX system — or any other system that deals with safety-critical information, such as patient clinical data — could jeopardize the patients' health if not solved. Therefore, we classify the CDX system interoperability as a safety-critical system.

As a safety-critical system, this interoperability inherits the safety concerns from the systems and adds up a whole new set of concerns. For example, if a test result

sent from a laboratory to a clinic does not arrive on time, intact and in a file format that the clinic's system can open, the clinical provider would not be able to make the diagnosis or treat the patient.

Hazard analysis is a suitable approach to improve the safety of a system. However, because interoperability of information systems often involves complex communication scenarios, performing the hazard analysis using traditional methods, such as FMEA or FTA, is a challenging task that may not achieve an optimal outcome. Therefore, a more system-oriented method, like STPA, is a better choice for the hazard analysis of this kind of system.

STPA is a method relatively easy to learn and simple to apply [58]; however, do it manually on paper or in a text document, especially for complex scenarios, can become a tangled task due to the lack of features such as traceability and input validation on those approaches. The STPA handbook [34] suggests limiting the number of system-level hazards analyzed to between 7 and 10 or grouping the hazards and refining them into sub-hazards. However, we argue that dealing with multiple documents manually is prone to errors and possible duplication or loss of information. Moreover, even limiting the number of system-level hazards, the analysis can lead to a large number of unsafe control actions and consequently many loss scenarios [56]. This scenario can be susceptible to errors or even be impractical to be performed manually without traceability tools.

Therefore, this thesis aims to answer the following research questions:

$RQ1$: How can we use or adapt hazard analysis methods to assure conformance profiles?

$RQ2$: How can we use or adapt hazard analysis tools to support this analysis?

1.4 Contribution

The main contribution of this work is the application of STPA in the CDX interoperability conformance profile with the scope of the eReferral clinical workflow in order to evaluate and assure the safety of the CDX interoperability. This work complements a previous study that considered only the ordering part of the eReferral workflow [66, 67]. Moreover, instead of using only word processing and diagraming software, we perform the hazard analysis using FASTEN, an open-source tool for specifying and verifying safety-critical systems. Furthermore, since FASTEN allows us to customize

the tooling and add features to support the analysis for our needs, we implement a set of new features in order to improve the FASTEN's support of STPA.

1.5 Outline

This thesis is organized as follows. Chapter 2 builds the background knowledge for this work by surveying interoperability for information systems, presenting the industry case CDX system utilized in the analysis, and reviewing the literature describing other works that applied STPA on information systems in the healthcare industry. Then, Chapter 3 discusses the method selection and adaptation by reviewing traditional and systemic hazard analysis methods, demonstrating the STPA method with a practical example, and describing the adaptations made to STPA for the analysis. Chapter 4 discusses the tooling selection and adaptation by reviewing and comparing open-source tools for STPA and describing the extensions made on FASTEN to help the analysis. Chapter 5 describes the application of STPA to evaluate the CDX interoperability conformance profile using FASTEN. Finally, Chapter 6 concludes this work by showing the results, limitations and future work.

Chapter 2

Background and Related Work

This chapter provides a non-systematic review of the interoperability studies focused on health information systems. Then, we introduce the Clinical Data eXchange (CDX) system, which is the interoperability industry case studied in this work. Finally, this chapter provides a survey of related works on hazard analysis.

2.1 Interoperability of Information Systems

Nowadays, for organizations to stay competitive and provide better products and services for their customers, they must cooperate with other business partners and other organizations at various levels [3]. Thus, new technologies and methodologies are increasingly fulfilling business cooperation at all levels to maintain business value [4]. For example, interoperability provides means of addressing business-to-business collaboration at different levels.

According to Gürdür and Asplund [19], interoperability has been studied for at least 40 years. Since then, various reference models and frameworks have been proposed. These proposals use levels or layers to perform interoperability assessment; hence they can be classified as maturity models, as observed by Guédria et al. [18]. A maturity model describes stages or levels of sophistication by which activities progress from the lower to the highest levels [4]. Thus, when systems or organizations reach higher levels of interoperability, it means an improvement in their capability to collaborate [7]. Moreover, as interoperability is closely related to the applied domain, each model is designed to satisfy its domain needs [16]. Furthermore, as the lower levels of interoperability are usually alike among domains, the maturity models are designed to

assess and improve the higher levels of interoperability, such as information systems, organizational and inter-organizational levels [19].

2.1.1 Levels of Interoperability

Interoperability has been studied since the 1970s, and, since then, a number of assessment models and frameworks have been proposed [16, 19]. These proposals usually employ levels or layers [62] and are designed as maturity models [16, 18]. As aforementioned, each model concentrates on a specific domain [16, 49]. For example, the European Telecommunication Standards Institute (ETSI) [62] lists four levels of interoperability — *technical*, *syntactical*, *semantic* and *organizational* — for networks and telecommunications. Benson and Grieve [6] show the four levels (or types) of interoperability — *technical*, *semantic*, *process* and *clinical* — defined by a healthcare organization. . The Healthcare Information and Management Systems Society (HIMSS) [20] defines four levels of interoperability — *foundational*, *structural*, *semantic* and *organizational* — for healthcare. Vernadat [63] applies three levels of interoperability — *technical*, *semantic* and *organizational* — to enterprise systems and networking. Moreover, the European Interoperability Framework (EIF) model has four layers of interoperability — *technical*, *semantic*, *organizational*, and *legal* — to provide electronic government services for business and citizens [15].

It is worth noting that all of these models are designed to address interoperability for specific domains. On the other hand, other interoperability assessment models aim to address information systems in general, independent of domain. Two examples of assessment models that focus on information systems are the Levels of Information Systems' Interoperability [7] and the Levels of Conceptual Interoperability Model [59]. Those are the most studied models in the specialized literature, according to Gürdür et al. [19].

Levels of Information Systems Interoperability

The Levels of Information Systems' Interoperability (LISI) [7] was created by the American Department of Defence (DoD), and it defines an *interoperability maturity model* to evaluate and improve interoperability of information systems throughout their lifetime. This maturity model is composed of five levels of interoperability:

L_0 *Isolated*, for stand-alone systems without direct connection;
L_1 *Connected*, for systems capable of transmitting data but not fusing information;

L_2 *Functional*, for distributed systems, locally connected and able to transfer data in formalized models;

L_3 *Domain-based*, for independent systems, remotely connected and able to share information through domain-based data models;

L_4 *Enterprise-based*, for systems spread across multiple domains and networks capable of dealing with advanced forms of collaboration.

In addition, LISI categorizes several aspects of interoperability for information systems in terms of four interrelated attributes:

- *Procedures*, which embraces development content and system architecture standards;
- *Applications*, which covers the system's functional requirements;
- *Infrastructure*, which supports the definition and utilization of the connection between systems;
- *Data*, which deals with the information processed by the systems.

LISI defines a *reference model* which describes the specific capabilities required to achieve each level of interoperability. Moreover, LISI refines this reference model into a more complex *capability model*, which defines some sub-levels of interoperability and more capabilities for each attribute. These three models (maturity model, reference model, and capability model) and other features form a complete interoperability assessment process to provide organizations with a reference framework for information systems interoperability.

Levels of Conceptual Interoperability Model

The Levels of Conceptual Interoperability Model (LCIM) [59] proposed by Tolk et al. aims to go beyond the scope of technical models for information systems interoperability, such as LISI, which focuses on the technical implementation of information systems. To achieve that, LCIM tries to establish a link between the technical implementation domain and the conceptual modelling domain. LCIM was initially defined with five levels of interoperability, similar to LISI. Then, it was embraced and improved by the community, hence being extended to seven levels of interoperability [60]:

L_0 *No Interoperability* occurs when there is no communication between systems;

L_1 *Technical Interoperability* occurs when exists a communication protocol that allows systems to communicate;

L_2 *Syntactic Interoperability* occurs when exists a shared data structure between systems;

L_3 *Semantic Interoperability* occurs when exists a semantic information model shared between systems;

L_4 *Pragmatic Interoperability* occurs when the systems are conscious of others' behaviours, and the information is contextualized.

L_5 *Dynamic Interoperability* occurs when the systems understand other systems states and are aware of changes in these systems' information.

L_6 *Conceptual Interoperability* occurs when there is a documented conceptual model allowing human interpretation and evaluation.

Even though the LCIM was conceived as a conceptual model for modelling and simulation systems [59], it seems that this model has been adopted as a reference model to address interoperability at information systems for various domains [60]. Moreover, Kubicek et al. [29] noticed that those models have the technical, semantic and operational interoperability levels in common, where information systems usually address the technical and semantic levels.

Technical interoperability is associated with how interoperability is accomplished. In other words, the necessary apparatus (communication protocols and infrastructure) for information systems transfer data. Thus, technical interoperability is achieved when services or information are successfully exchanged between systems [49]. However, technical interoperability only guarantees that the data are transmitted. It does not say the format or syntax of the transferred data [29].

As can be observed in the LCIM, a syntactical level is defined between technical and semantic levels. This Syntactic interoperability is responsible for defining the data format or syntax utilized to transfer data between systems. Thus, to achieve syntactic interoperability, there must be an agreement on the syntax and encoding of the exchanged data [62]. However, many models do not define the syntactic level, so its concerns and objectives are accumulated and addressed by the previous level, which is the technical interoperability.

The other level typical for information systems is semantic interoperability. This level is associated with the data content and is generally related to human interpretation and analysis [49]. For information systems to interoperate at this level, they have to "understand the meaning" of the exchanged information [59]. Due to this "human behaviour", semantic interoperability was considered one of the most critical

challenges for heterogeneous environments in the earlier 2000s [43] and is still one of the most discussed topics in the related literature [49].

Semantic interoperability for information systems is associated with different systems' ability to exchange and interpret information [49]. Several studies focus on semantic interoperability, trying to give information systems the ability to interpret data, or in other words, generate and consume relevant information [29].

Park and Ram [43] categorize semantic interoperability into three general areas, which should be mixed to achieve interoperability:

- *Mapping-based approach* relies on mapping or schema definitions to connect multiple sources;
- *Intermediary-based approach* utilizes an intermediary structure to build the link between distinct sources;
- *Query-oriented approach* is based on declarative or structured query languages to search for related information on specific data sets.

Different from the previous categories, which group the mechanisms utilized to share information between systems, Chen et al. [12] categorize the interoperability approaches focusing on how different information are merged:

- *Integrated approach* when the parts agree on a common format;
- *Unified approach* when exists a common meta-model to be mapped by each part;
- *Federated approach* when does not exist an imposed model, but the participants share ontologies.

2.2 Health Interoperability Standards

Various industries have organized groups to define standard models and provide reference frameworks to enable legacy and new systems to interoperate within their domains [52]. For instance, in the healthcare industry, over 40 different organizations are working on developing, maintaining, and implementing standards for health information [20].

Health Level Seven (HL7) International is a global organization formed by healthcare experts, government and industry stakeholders, and others collaborating to create and provide a comprehensive framework and standards for exchanging, integrating, and managing electronic health information [22]. The HL7 standards are

designed based on generally agreed common requirements across healthcare organizations worldwide, and hence these standards are widely adopted. For example, Canada Health Infoway [53] (Infoway) is an organization affiliated with HL7 International that holds a number of responsibilities within HL7 International. Some examples of these responsibilities are: promote and support HL7 related pan-Canadian health information standards; and promote and support domestic and international HL7 agreement to the benefit of health information standards in Canada.

2.3 Clinical Data eXchange

Clinical Data eXchange (CDX) is a clinical document distribution service build on top of HL7 standards. CDX is conceived by the health authorities Interior Health[1] and Northern Health[2] from British Columbia, Canada. The main goal of CDX is to facilitate the exchange of clinical documents between health authorities and health providers' EMR systems and among EMR systems across the province [27].

The clinical documents manipulated by the CDX system are standardized XML documents based on the HL7v3, known as Clinical Document Architecture (CDA). A CDA document is formed by a header and a body, where the header identifies the document and holds the patient, author and recipients' information, and the body contains the document content or clinical report [13]. The CDA document body varies according to the document's conformance level. Figure 2.1 shows a visual representation of the CDA document Level 1, described below. This Figure only illustrates components essential for the hazard analysis and thus omits most of the CDA elements. Notice that the attachments are not part of the CDA document but are transported side-by-side in the transmission wrapper defined by the standard.

[1] `https://www.interiorhealth.ca/`
[2] `https://www.northernhealth.ca/`

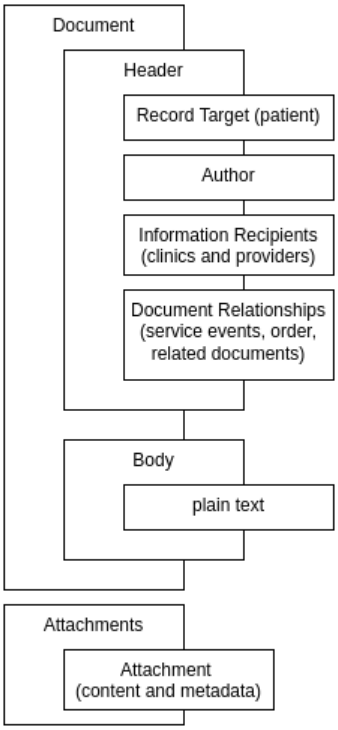

Figure 2.1: Visual representation of the CDA structure. Adapted from the CDA Implementation Guide [36]

2.3.1 CDA Conformance Levels

CDX defines three conformance levels for CDA that increase the complexity of the documents and hence their interoperability capabilities [50].

- CDA Level 1 is comprised of an XML header and an unstructured text body.
- CDA Level 2 contains an XML body with defined sections and the XML header of level 1. These sections represent narrative blocks and are specified by templates.
- CDA Level 3 includes discrete data elements in the XML body sections of level 2. These data elements can be referenced to relevant industry-standard code systems.

All three CDA levels varies between the syntactic and semantic levels of interoperability, depending on which part of the document we observe. The XML header, for instance, is at the semantic level for all cases because the information in the XML head has meaning and is utilized by all systems involved in the communication.

The CDX system uses the recipient information (clinics and providers) to route the documents to the correct clinics, and the EMR systems use the target (patient information), recipients (providers) and documents references to manage the documents within the system.

On the other hand, the XML body of CDA level 1 is in the syntactic level of interoperability in most cases because it is essentially plain text only, which has meaning only for the humans reading it. However, for CDA Levels 2 and 3, the XML body starts to exhibit defined sections that can be read and interpreted by computing systems more clearly. Thus, it moves to the semantic level of interoperability.

In addition, the CDX system is apt to receive and process documents at all three levels. However, the CDX Conformance Profile [50] aims at the CDA Level 1. Moreover, EMR systems are expected to receive and render documents on levels 2 and 3 at CDA Level 1, with the exception of CDA Laboratory Results, which is out of the scope of this work.

2.3.2 CDX System Architecture

The CDX system is a centralized distribution system that receives and distributes clinical documents asynchronously. It utilizes SOAP Web Services with Web Service Security over HTTPS to implement the system's services. The services provided by CDX are organized into document submission, document retrieval and organizational queries [11], and each set provide the following services.

The document submission consists of the *CDA Submit* service, responsible for receiving and validating documents. Since the CDX document distribution is asynchronous, this service responds to the document submission with an acknowledgment indicating that the CDX system has received the document and the result of the validations performed on the document on arrival. Moreover, CDX implements the *Distribution Status* service, which allows a client system to keep track of the status of the submitted documents.

For document retrieval, CDX implements multiple services. First, the *List New Documents* service returns a list of document metadata for the documents that the requesting clinic did not retrieve yet. Then, the *Search Documents* service uses search criteria to return a list of document metadata. Finally, the *Get Document* service returns the content of the requested document. It is worth mentioning that the documents' data and metadata returned by these services are also filtered by receiving

clinic, i.e. only the recipient is allowed to retrieve the data.

Lastly, the organizational queries set is composed of services that retrieve locations and providers registered within CDX. For instance, The *Search Locations* service returns the details of the locations that can receive documents, and the *Search Provider* service returns the details of the providers and locations that each provider is associated with. When included in the submitted documents, this information is utilized by the CDX system to address the document to the correct recipients.

2.4 Related Work

The works listed here were collected from the publications available on the MIT Partnership for Systems Approaches to Safety and Security website [3] and searches in academic research databases.

Samost applies STPA as a proof of concept on a novel radiation oncology process, showing positive and promising theoretical results of the hazard analysis using STPA compared to techniques grounded in a linear chain of events, such as FMEA [51]. The author indicates that STPA can benefit the healthcare industry because this industry usually employs complex socio-technical systems and processes.

Pawlicki et al. apply STPA to radiation oncology from the clinical perspective [44]. They detected eighty-three UCAS and 472 loss scenarios in their analysis. The authors also performed the hazard analysis of the same use case using FMEA and compared the outcomes of both methods, pointing out similarities and differences between the two methodologies. Overall, the two techniques do not produce the same result since each method is formulated on different theories — reliability theory for FMEA and systems theory for STPA. Furthermore, they emphasize that a number of factors make it challenging to validate a hazard analysis technique's completeness, and hence determining which is the best approach is not possible.

Mason-Blakley presents a systemic hazard analysis technique for clinical information systems based on STAMP, other hazard analysis methods, and industry safety standards [38]. Mason-Blakley also mapped a set of patterns to support the application of the developed technique.

Mindermann et al. performed an exploratory study on the extension of STPA for privacy (STPA-Priv) to evaluate complex privacy risks of an eHealth scenario

[3] `http://psas.scripts.mit.edu/home/materials/`

involving a smart glucose measurement device for children [40]. The studied scenario involves users' private and sensitive information that can be shared with third parties besides the patients' physicians, challenging users' privacy. They highlight that the application of the STPA-Priv is intuitive and raises many privacy concerns that could be implemented to improve the system. Furthermore, they use the open-source tool XSTAMPP to support the analysis. However, although the toll is helpful and facilitates things like modelling the system and organizing the analysis artifacts, the lack of documentation creates extra challenges for the study.

Kaberuka and Johnson adapted the extension of STPA for security (STPA-sec) for a socio-technical cyber security scenario using a risk management system for health care in emerging nations as a case study [28]. The study concludes that the methodological adaptations improve cyber security and risk management in the institution where the study was conducted. In addition, the study raised a number of questions related to reproducibility and acceptance of their methodologies in other challenging scenarios.

Silvis-Cividjiana et al. use STPA to analyze the safety of a radiation therapy process [54]. They compare the STPA outcomes to an FMEA study on the same therapy process. The study highlight that STPA requires a mindset change to complete the modelling part of the analysis. However, when the analysis purpose and control structure are done, they facilitate the detection of hazards. They explain that FMEA can identify more detailed and domain-related hazards, while STPA is more accurate in describing unsafe scenarios, including uncharted ones. Furthermore, they noticed that the STPA methodology leads to valuable insights to improve system safety and clarify complex socio-technical workflows.

Bas adapted STPA to a system for monitoring and tracking diabetes mellitus [5]. Bas proposes including a consensus-based process in the STPA steps, systematic generating the UCAs, and analyzing the controllers' interactions. Bas also provides a non-systematic literature review with other works that address STPA in healthcare.

In summary, STPA is a recent technique for performing hazard analysis and is still not accepted widely by the industry as other techniques, such as FMEA or FTA. However, both academy and industry have shown more interest in STPA as an alternative, or complement, to traditional hazard analysis methods.

Chapter 3

Method Selection and Adaptation

In order to help answer the RQ1 — *How can we use or adapt hazard analysis methods to assure conformance profiles?* — we surveyed hazard analysis methods present in the current literature; and compare these methods using a set of requirements to address the analysis of conformance profiles. Then, we give an overview of the System Theoretic Process Analysis (STPA) method and present the proposed extension of STPA for the analysis of conformance profiles.

3.1 Hazard Analysis Methods Survey

Over 100 hazard analysis techniques are described in the literature, many of which are variations and adaptations of more widespread methodologies, and many are not broadly disseminated [14]. The most common methodologies utilized nowadays were coined between the 1950s and 1970s. Those methodologies were conceived considering that failures only happen on physical components, and those failures were usually the result of natural degradation, inadequate maintenance, operator malpractice, environmental disturbances, or other physical-related issues [32]. However, the growing complexity of socio-technical systems introduced new challenges for those traditional methods, and new approaches emerged as a result [23].

Hazard analysis can be applied to all phases of a project, from the concept development to the operation and maintenance [34]. For example, if applied during the concept development or requirement engineering phases, hazard analysis methods can generate system requirements. For instance, some industries, such as aerospace, automotive, medical, and nuclear, determine that safety-critical systems are analyzed

in the early stages of their projects.

Although this work studies the application of hazard analysis methods for conformance profiles, which are usually employed to conduct system tests and evaluations, conformance profiles are directly associated with the system specification. Thus, they deal with system concepts and functions instead of components. In addition, generating testing profiles or even evaluating third-party conformance profiles using hazard analysis can ensure that critical tests cover safety-critical parts of the system. So, with a focus on testing compliance profiles, an ideal hazard analysis method should:

- perform a prospective hazard analysis;
- be suitable for complex socio-technical systems;
- be appropriate for requirements generation; and
- deal with different types of system documentation.

We narrow our survey to the most commonly applied traditional methods [33], namely *Failure Mode and Effects Analysis (FMEA*), *Fault Tree Analysis (FTA)*, *Event Tree Analysis (ETA)*, and *Hazard and Operability Analysis (HAZOP)*, as well as some well-known systemic methods for analysing socio-technical systems [61], specifically *Functional Resonance Analysis Method (FRAM)* and *System Theoretic Accidents Models and Processes (STAMP)*.

3.1.1 Failure Mode and Effects Analysis

Failure Mode and Effects Analysis (FMEA) was first released in 1949 as a military standard for system reliability and performance. It is still a prevalent technique, considering all the variations across different industries [14]. Moreover, despite FMEA being created initially for reliability engineering, it was extended for other areas, such as hazard analysis [14, 37].

FMEA is a bottom-up hazard analysis that uses a linear chain of events technique to analyze each step of the process (or system) for potential failures. The analysis starts by identifying all components of the system and its operational modes. Next, the failure modes of each component are identified, together with their causes and how they can be detected. Then, the effects of the failures are analyzed. This part changes according to the area of the analysis. For hazard analysis, we are interested in the resulting hazards and accident risk [14, 37]. Next, the failure rates and severity for each failure mode are calculated.

Moreover, it is possible to perform a quantitative analysis if the component failure rates are available; however, the quantitative analysis is not possible for software systems because failure rates are not available [37]. In addition, FMEA can be performed on three different approaches: functional, physical, or hybrid. For instance, the functional approach is more suitable for software systems because it focuses on the system's functions instead of its components [14].

FMEA is an easy and inexpensive technique to perform, yet rigorous and efficient method for analyzing components reliability and detecting single component failures on hardware and process-oriented systems [37]. However, because FMEA focuses on single-initiated failures, it is not suitable for the complexity of socio-technical systems. Moreover, because FMEA focuses on the direct cause and effects of potential failures, the analysis is limited to events occurring near the accident and hence not suitable for the complexity of socio-technical systems [14]. Furthermore, another restriction of FMEA is that the Risk Priority Number (RPN), utilized to assess the hazards quantitatively, is a semi-quantitative risk measurement, and it requires external information, often provided for experts' knowledge instead of empirical data [58].

3.1.2 Fault Tree Analysis

Fault Tree Analysis (FTA) was created at Bell Labs in 1961 to analyze a missile guidance system, and then it was extended by Boeing to other military systems [14]. Due to its successful applications and analytical capabilities, FTA quickly spread to other areas dealing with complex dynamic systems, such as aerospace and nuclear power industries [30].

FTA is a top-down hazard analysis technique that employs a fault tree diagram containing all the system events that lead to a top-level hazardous or undesirable event under investigation. The fault tree is a structured logic diagram that uses logic gates and event symbols to represent the system design from a failure state perspective. The analysis begins with the statement of the hazardous event and the definition of the system, including the system failure modes and boundary conditions of the analysis. Then, using this information, the fault tree diagram is constructed using various modelling techniques [30]. Finally, with the diagram complete, the analysis proceeds to evaluate the fault tree.

The fault tree evaluation can be qualitative or quantitative, depending on the

scope and output of the analysis. For instance, qualitative analysis is going to determine the minimal cut set or path set of the fault tree that leads to the undesirable event. This minimal cut set helps in the prioritization of component failures and reduction of engineering efforts. On the other hand, quantitative analysis can help calculate the probability of occurrence of an undesirable event. Nonetheless, quantitative analysis depends on the availability of more information, such as component failure rate and fault duration of all components under investigation.

FTA is a well-accepted hazard analysis technique, especially in industries where the systems have well-known failure rates, modes, causes and effects. Moreover, FTA is a robust and efficient method for identifying and preventing single component failure. However, it is not adequate to identify failures provoked by component interaction, especially interactions between non-faulty components, or failures caused by design flaws or flawed specification [34]. Furthermore, FTA lacks systematic mechanisms for capturing failures originated from human behaviour and software systems, both frequent on socio-technical systems [58].

3.1.3 Event Tree Analysis

The Event Tree Analysis (ETA) method was created in 1974 during the safety analysis of a complex power plant as a simplified alternative to FTA while still using an events-based methodology [14].

ETA is a bottom-up hazard analysis that uses an event tree diagram to evaluate the possible outcomes, or accident scenarios, of a sequence of events. The analysis begins by identifying the initiating failure event, and the pivotal events intended to prevent an undesirable state of the system. Then, the binary tree is drawn from the initial event and branching on each pivotal event, tracing then success and failures paths until they reach the possible outcomes. With the event tree diagram ready, the quantitative analysis can be performed if the likelihoods of the events are known.

ETA is an easy-to-follow structured and systematic approach to assessing the probability of single failure events to occur given preventive measures (pivotal events). However, ETA is designed for single events only, and like FTA, ETA is not appropriate to analyze component iterations or design flaws. Moreover, ETA is not suitable for systems where events result in non-binary outputs, like human behaviour [14]. Furthermore, the lack of systematic methodology for identifying initiating events make ETA dependent on other hazard analysis technique [58].

3.1.4 Hazard and Operability Analysis

Hazard and Operability Analysis (HAZOP) was developed in the 1970s by the chemical industry and later embraced by the petroleum, food, water, and other industries [14].

HAZOP is a very structured and methodical process for hazard identification. The method uses a set of parameters, guide words and system diagrams to help the identification of hazards resulting from potential operational deviations. Even though HAZOP was developed for process design and operations, it can be extended to systems and functions. Moreover, HAZOP is usually carried out by a multidisciplinary team with experience in the system's design; Furthermore, HAZOP can be performed at different levels of abstraction, such as conceptual, top-level, and components.

The analysis begins by defining the scope and boundaries of the system, including system mission, mission phases and environments; planning the analysis; selecting the team; and acquiring the relevant data of the system. Next, the analysis is conducted by identifying the items to be evaluated, establishing the appropriate system parameters, guide words, and worksheets, conducting the meetings, recording and validating the results. During the meetings, the analysts compare a list of system parameters against a list of guided words with the objective of identifying possible deviations in the system design that can lead to hazards. Finally, when the analysis is concluded, HAZOP predicts further steps for applying and monitoring the corrective actions.

HAZOP is a well-defined and easy-to-apply technique with great potential to capture accidents from component failures. However, it requires a very detailed system design and can be time and labour intense [14, 58]. Moreover, like other techniques not designed for software systems, HAZOP focuses on single events and hence is not a well-suit for identifying hazards originated from multiple events or interaction between events [14].

3.1.5 Functional Resonance Analysis Method

Functional Resonance Analysis Method (FRAM) is a hazard and accident analysis method proposed by Erik Hollnagel in 2004 [24].

FRAM uses the analogy of resonance as an alternative to traditional methods based on cause-effect relations. It considers that accidents may occur as a result of conditions created by the resonance of the variability in the system functions. The FRAM analysis is conducted with the assistance of a diagram that represents the

functions of the system and their relationship. The diagram is formed by hexagons that represent the system functions, and each vertex of the hexagon represents a relationship attribute, defined as follows:

- *Inputs* required to execute the function
- *Outputs* of the function
- *Resources* needed by the function to process the inputs
- *Controls* that manage or bound the function
- *Preconditions* that must be fulfilled before the function execution
- *Time* of the function operation and allowable time window for the function to execute

The analysis begins by identifying the essential system functions and describing how they are usually carried out using the six FRAM relationship attributes. Next, the potential for variability of each function is calculated using the performance criteria of the system domain. Then, the analysis looks for the possibility for functional resonance, which happens when the variability of functions interact or are combined, so a specific function is incorrectly executed. Finally, the analysis proposes ways to address uncontrolled performance variability or conditions of functional resonance and prevent them from causing an accident.

FRAM specifies a system representation that contributes that the analysts are on the same page regarding the system functions and assumptions utilized in the analysis. Moreover, the functional representation emphasizes functions interactions and dependencies. However, as FRAM is a new technique, it lacks studies, particularly involving information systems [58].

3.1.6 System Theoretic Accidents Models and Processes

System Theoretic Accidents Models and Processes (STAMP) is a methodology for system safety, created by Nancy Levenson in 2002 [31]. STAMP is based on system theory and treats safety as a control problem, in which complex systems are viewed as a hierarchical controller structure formed by control loops.

Control loops are composed of *controllers*, *sensors*, *actuators*, and *processes*. Controllers have a *process model* to maintain the state of the controlled process and a *control algorithm* to make decisions. Within a control loop, a controller sends *control actions* to a controlled process via an actuator; and receives *feedback* from the

process through a sensor. Moreover, the controller can receive external inputs and send status to other elements in the control structure; similarly, the controlled process can receive inputs and send outputs to other parts of the control structure. Any issue on the control loop, such as component failure, missing or incorrect feedback, or inadequate control action, can lead to an accident.

STAMP can be utilized to perform hazard analysis and accident analysis, using STAMP-based methods STPA (System Theoretic Process Analysis) and CAST (Causal Analysis based on Systems Theory), respectively. For hazard analysis, STPA starts by defining the purpose of the analysis and modelling a functional control structure diagram for the system under investigation. These first steps define the purpose of the analysis and delimit the system's boundaries. Then, using the control structure, the analysis focuses on identifying the unsafe control actions (UCA) that can lead to pre-defined hazards. Then, STPA aims on explaining why UCAs might occur by determining potential loss scenarios that could lead to UCAs and scenarios that lead to hazards without UCAs. This process is explained in detail in Section 3.2.

Some works indicate that STAMP detects fewer hazards than traditional approaches for component failure hazards; it also does not prescribe any criteria for validation or prioritization of risk for quantitative risk assessment [57, 70]. In addition, systemic analysis models are still new compared to traditional ones and do not have a large number of industry cases to back them up [69].

3.1.7 Hazard Analysis Summary

Traditional methods, namely FMEA, FTA, ETA, and HAZOP, are based on a linear-event approach, which is not appropriate for hazard analysis of complex socio-technical systems [24, 33]. Some of the limitations of this type of method include:

- The system events are predictable and sequential;
- The system components and functions are discrete, e.g. work or fail;
- The accidents arise from a sole root cause;
- The human operators are the first to be blamed for accidents;

On the other hand, systemic methods also have some caveats [45, 24, 33], including:

- They can be time and resource-consuming;
- Technical expertise is required;

- They only offer qualitative analysis;

Table 3.1 summarizes the hazard analysis methods presented above, classifying each of them according to the following characteristics.

- The *paradigm* of the hazard analysis method. E.g. event-based, component-based or system-based.
- *Quantitative or qualitative* analysis.
- The *reasoning* that best encompasses the method. It could be deductive reasoning, in which the conclusion is drawn from a defined set of premises, or inductive reasoning, which proposes a conclusion based on more information than the observed.
- *Top-down or bottom-up* approaches.
- The *Hazard identification method*
- The method *linearity*. E.g. linear or systemic.

After analyzing the characteristics of the methods and considering their strengths and caveats, we select the STPA to perform the hazard analysis of conformance profiles. In summary, STPA is a prospective hazard analysis method; it is designed for analyzing complex socio-technical systems; and is appropriate for requirements generation, as stated by many studies [58, 44, 64, 70].

Method	**Paradigm**	**Quant. / Quali.**	**Reasoning**	**Top-down / Bottom-up**	**Hazard ident. method**	**Linearity**
FMEA	component-based	quant. or quali.	inductive	bottom-up	type of functional failures	linear
FTA	event-driven	quant. or quali.	deductive	top-down	fault tree	linear
ETA	event-driven	quant. or qual.	deductive	bottom-up	binary tree	linear
HAZOP	system-based	quali	inductive	bottom-up	system parameters & guide words	linear
FRAM	system-based	quali	inductive	top-down	aggregation of variability in functions	systemic
STAMP / STPA	system-based	quali	inductive	top-down	types of UCAs	systemic

Table 3.1: Summary of Hazard Analysis Methods

3.2 System Theoretic Process Analysis

STPA is a hazard analysis method based on the Systems-Theoretic Accident Model and Processes (STAMP) initially developed by Leveson [31]. STPA perceives safety as a control problem. Therefore, the system under analysis is modeled using a set of control loops named *control structure*. This control structure is a functional control diagram that can include elements from the entire process in analysis, such as system's functions, organizational and operational decisions.

STPA can be applied during any phase of the system life cycle, including early in the system designing when STPA will provide safety requirements and constraints for

the system development's subsequent phases [33]. Like any decision that could result in changes in a system's design, applying STPA during the system design phase is more cost-effective for the project.

STPA has four main steps: (1) define the purpose of the analysis, (2) model the control structure, (3) identify UCAs and (4) identify loss scenarios [34]. These steps are explained below with the support of a simple example.

3.2.1 Step 1: Define the purpose of the analysis

The first step of STPA consists of defining the purpose of the analysis. This is done by identifying the *losses* the analysis aims to prevent, and the *system-level hazards* with corresponding *system-level constraints*.

It is essential to establish the system boundaries for the analysis in order to perform this step correctly, especially for the identification of system-level hazards. As defined in Chapter 1, a *system-level hazard* is a hazard delimited by the system's boundaries. Moreover, for purposes of the hazard analysis, the system is an abstraction of the actual system. This abstraction is defined by the analysts at this stage of the analysis. In addition, a *system-level constraint* is a system condition or behaviour intended to hinder one or more hazards and, consequently, prevent losses [34].

Identifying losses

As an example, consider an automatic door system for elevators. In this example, the boundaries are defined by the functions the automatic door system performs and how it interacts with the environment, i.e. the system's inputs and outputs. The losses to be considered for this analysis are about the safety of the people using the elevator.

$L1$: A person is harmed by being hit by a closing door
$L2$: A person is harmed by falling in or hitting the elevator shaft
$L3$: A person is trapped inside the elevator during an emergency

It is worth mentioning that other emergent properties could be considered when choosing the losses for the analysis. STPA can address any loss that is unacceptable to the system stakeholders.

Identifying system-level hazards

After identifying the losses, the system-level hazards that could lead to these losses are identified. Each hazard should describe a system state or condition that can lead

to one or more losses. Moreover, hazards can be written using the structure: *Hazard = <System> <Unsafe condition> [Link to Losses]*

These are the hazards that could lead to losses previously identified for the automatic door system:

$H1$: Doors close on a person entering or exiting the elevator [L1]
$H2$: Doors open when the elevator is moving or is not stopped on a floor [L2]
$H3$: Closed doors prevent people to exit during an emergency [L3]

Defining system-level constraints

System safety constraints must ensure that system-level hazards do not happen. Therefore, a system safety constraint is usually the inverted condition of a hazard. Like the hazards are linked to one or more losses, the safety constraints must be linked to one or many hazards. Moreover, safety constraints can be written using the structure: *Safety Constraint = <System> <Condition to enforce> [Link to Hazards]*

The safety constraints that prevent the previously identified hazards from happening are as follows:

$SC1$: Doors must not close when something is in the doorway [H1]
$SC2$: Doors must not open when the elevator is moving or is not stopped on a floor [H2]
$SC3$: Doors must not close during an emergency [H3]

3.2.2 Step 2: Model the Control Structure

The second step of STPA is to model the system in terms of control loops, forming a hierarchical control structure that helps perform the rest of the analysis. A simplified control loop is composed of a *controller* that provides *control actions* to control a process and thus enforcing constraints on this *controlled process*. The control loop also has a *feedback*, which the controller uses to observe the controlled process. Moreover, the control loop can interface with components beyond its boundaries. Inputs and outputs are added in the diagram to represent these interfaces.

Additionally, the control loop can have components representing the *actuators* and *sensors* between the controller and the controlled process. The actuators expose how the control actions are executed, and the sensors demonstrate how the feedback is detected.

Figure 3.1 shows the control structure for the automatic door system.

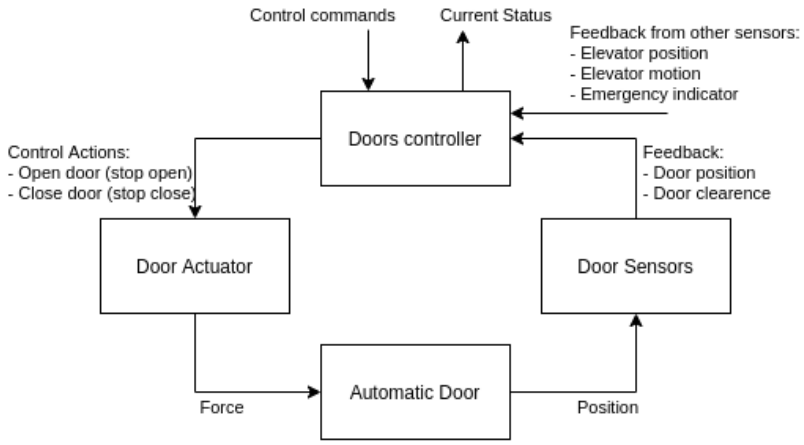

Figure 3.1: Control Structure for Automatic Door System

3.2.3 Step 3: Identify Unsafe Control Actions

The third step of STPA looks at the control structure and identifies potentially unsafe control actions (UCAs) that can lead to one or more of the previously recognized hazards. STPA defines four categories of UCAs that can lead to one or more hazards.

- A safe control action that is not provided
- An UCA that is provided
- A safe control action that is provided too late, too early, or out of sequence
- A safe control action that is stopped too soon or applied too long

Table 3.2 shows the UCAs for the automatic door system. UCAs can be written using the structure:

UCA = <Source> <Type> <Control Action> <Context> [Link to Hazards]

where *<Source>* is usually the controller, *<Type>* is the category of the control action, *<Control Action>* is the control action itself, and *<Context>* is the context where the hazard can happen.

As can be noticed the table, the UCA1, UCA6, and UCA7 do not cause any hazard previously defined; therefore, these UCAs do not need to be further analyzed. Moreover, UCA3, UCA4, and UCA5 cover the same case and can be rewritten as only one UCA, e.g., "UCA3: Doors controller provides 'open door' when the elevator is not stopped on the floor correctly [H2]".

Control Action	**Not provided**	**Provided**	**Too early, too late, out of sequence**	**Stopped too soon, applied too long**
Open door	UCA1: Doors controller does not provide 'open door' when the elevator stops on a floor [not a hazard] UCA2: Doors controller does not provide 'open door' while doors are closing and something is in the doorway [H1]	UCA3: Doors controller provides 'open door' while the elevator is moving [H2] UCA4: Doors controller provides 'open door' when the elevator stops in the wrong position [H2] (similar UCA3)	UCA5: Doors controller provides 'open door' too early before the elevator stops on a floor [H2] (similar UCA3) UCA6: Doors controller provides 'open door' too late after the elevator stops on a floor [not a hazard]	UCA7: Doors controller stops providing 'open door' too soon, not opening door completely [not a hazard]
Close door	UCA8: Doors controller does not provide 'close door' before the elevator starts moving [H2]	UCA9: Doors controller provides 'close door' while something is in the doorway [H1] UCA10: Doors controller provides 'close door' during an emergency [H3]	UCA11: Doors controller provides 'close door' too late after the elevator starts moving [H2]	UCA12: Doors controller stops providing 'close door' too soon, not closing door completely [H2]

Table 3.2: UCAs for the automated doors control system

At this point of the analysis, STPA advises that the UCAs identified during this

step are utilized to create safety requirements and constraints for the system.

3.2.4 Step 4: Identify Loss Scenarios

In this step last of STPA, each UCA is examined in conjunction with the control structure to identify the loss scenarios. The loss scenarios can be divided into two categories:

1. Loss scenarios that lead to UCAs
2. Loss scenarios in which control actions are improperly executed or not executed

Each category tends to affect a specific part of the analysis. For instance, Figure 3.2 shows where each loss scenario category acts on the control loop.

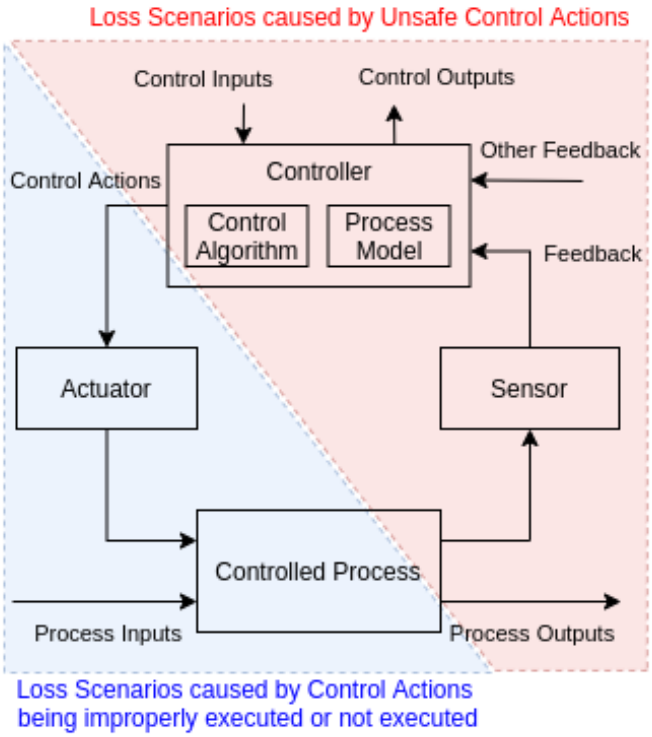

Figure 3.2: Loss Scenarios categories. Adapted from the STPA Handbook [34]

Scenarios that lead to unsafe control actions

Loss scenarios that lead to unsafe control actions are identified by going backwards in the STPA steps, i.e., starting from the unsafe control action. The main goal is

to explain what could cause the controller to provide or not provide the UCA. The main factors to consider in this type of scenario are unsafe controller behaviour and inadequate feedback. Figure 3.3 shows where these factors actuate in a generic control loop.

An *unsafe controller behaviour* may be caused by failures involving the (physical) controller, inadequate control algorithm, unsafe control input, or inadequate process model. Furthermore, *inadequate feedback* may be induced by missing feedback or wrong feedback received.

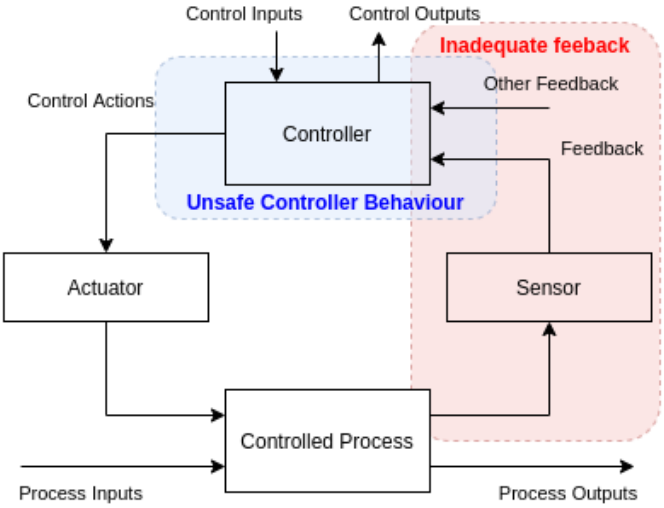

Figure 3.3: Factors that lead to UCAs. Adapted from the STPA Handbook [34]

Unsafe controller behaviour

Loss scenarios for *failures on the physical controller* are created stating with the UCA, thus identifying the controller and the physical failure that explains it. For example:

> **UCA2**: Doors controller does not provide 'open door' while doors are closing and something is in the doorway [H1].
> **Scenario**: Doors controller receives the feedback that something is in the doorway while the door is closing; however, due to an internal failure, the doors controller does not provide the open door action. As a result, the doors close on a person entering or exiting the elevator [H1].

Loss scenarios caused by an *inadequate control algorithm* are created starting with the UCA and identifying how the control algorithm may cause the UCA. For example:

UCA11: Doors controller provides 'close door' too late after the elevator starts moving [H2].
Scenario: Doors controller receives the command to close the doors, but a processing delay within the controller results in the close door control action being provided too late [UCA11]. As a result, the doors keep opened while the elevator is moving [H2].

Loss scenarios related to *unsafe control inputs* are created by observing other controllers that iterate to the controller under analysis.

Loss scenarios caused by an *inadequate process model* are created starting with the UCA and identifying the controller process model (beliefs) that can cause the UCA. For example:

UCA10: Doors controller provides 'close door' during an emergency [H3].
Process model (belief): Doors controller believes the elevator is stopped in an improper position.
Missing or invalid feedback: Emergency indicator is not received or is ignored by doors controller.
Scenario: During an emergency, the doors are opened manually by fire-fighters or trained personnel to help the people trapped inside the elevator exit it safely. Doors controller provides 'close door' because the elevator is stopped in an improper position. As a result, the rescuers' work is disrupted, and the people trapped inside the elevator are prevented from exiting safely [H3].

Inadequate feedback

Loss scenarios involving *inadequate feedback* are created by identifying the UCA and the feedback responsible for the UCA. For example:

UCA3: Doors controller provides 'open door' when the elevator is not stopped on the floor correctly [H2].
Scenario: Doors controller receives incorrect feedback about the elevator motion or position, opening the doors when the elevator is in the improper position or moving [H2].

Scenarios in which control actions are improperly executed or not executed

Loss scenarios caused by *control actions being improperly executed or not executed* do not depend on UCAs. These scenarios can occur in the control path and controlled process, as shown in Figure 3.4. For instance, the control path goes from the controller to the controlled process. Moreover, the factors that trigger these scenarios prevent the control action from being applied to/received by/executed by the controlled process.

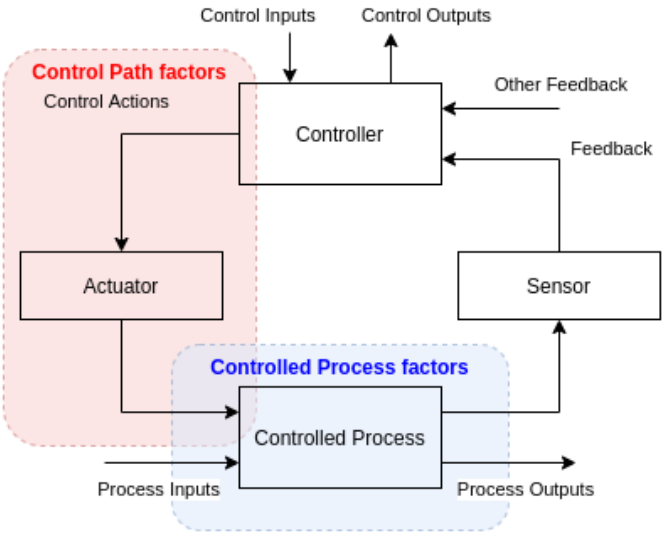

Figure 3.4: Factors that cause improper control actions. Adapted from the STPA Handbook [34]

Control path factors

Scenarios involving the *control path* are created, starting with the control action, and thus identifying what is being executed improperly or not executed, and identifying how the control path contributes to that behaviour. For example:

> **Control action**: Open door.
> **Improper/No execution**: Doors are not opened.
> **Scenario**: Doors controller sends open door while doors are closing upon receiving the feedback that there is an object in the doorway, but the door is not opened due to a failure on the door actuator. As a result, the doors close on a person entering or exiting the elevator [H1].

Controlled Process factors

Scenarios related to the *controlled process* are created by selecting a control action and identifying what factors can affect the controlled process, making the control action ineffective.

> **Control action**: Close door.
> **Scenario**: Doors controller sends close door, but the doors are not closed entirely due to some debris in the door's tracks that also trigger the doors sensors informing the doors are closed. As a result, the elevator starts to move with the doors opened [H2].

3.3 Extending STPA

As aforementioned, STPA is a prospective hazard analysis method suitable for investigating complex socio-technical systems. In addition, STPA can be applied to the whole project life cycle, including concept development, requirements engineering, system test, and other project phases [34].

This study performs the hazard analysis on the interoperability conformance profile for the CDX system, described in Section 2.3. Conformance profiles are guidelines for conformance testing created from the system specification. In order to better accommodate both system specification and conformance profiles in the analysis, we extended the STPA method as outlined in Figure 3.5 and explained in the following sections.

The main differences from the original STPA method are two dedicated steps to identify controller constraints and an additional step to align constraints and identify missing or incorrect ones.

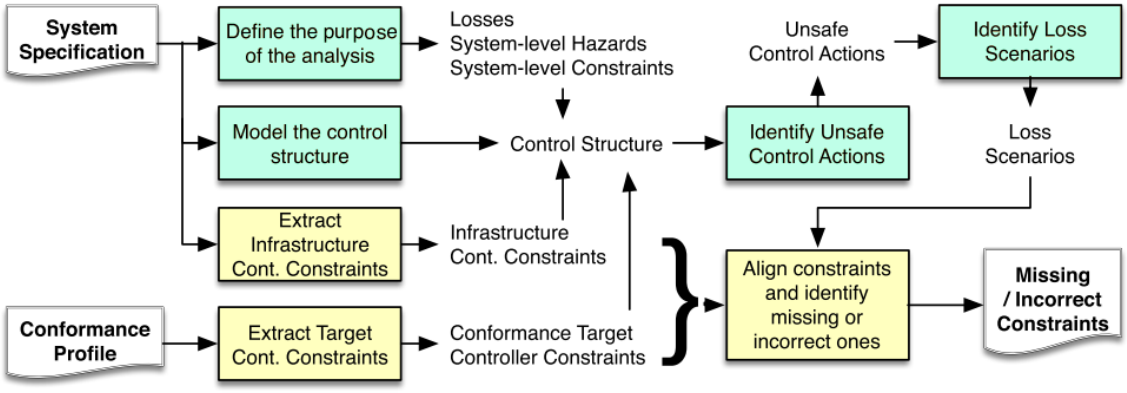

Figure 3.5: STPA-based hazard analysis method for interoperability conformance profile (Cyan boxes identify the original STPA method, and Yellow boxes identify the extensions) [67]

3.3.1 Extracting Controller Constraints

The analysis takes two inputs, the (Overall) *System Specification* and the *Conformance Profile*. It is worth mentioning that these two documents can be part of the same physical document that would have a specific section with the conformance testing guideline.

The System Specification is utilized to the STPA steps 1 and 2 as described in Section 3.2 and the new step that extracts the *Infrastructure Controller Constraint*. Then, the Conformance Profile is utilized as an input to identify the safety-related constraints (*Conformance Target Controller Constraints*). Identifying these last constraints is straightforward because conformance profiles tend to be highly structured documents. On the other hand, identifying infrastructure constraints could be more challenging, as overall system specification tends to be descriptive system explanations instead of prescriptive requirements specifications.

3.3.2 Aligning Constraints

After extracting the controller constraints from the documents, the standard STPA steps 3 and 4 are performed. At this point of the analysis, controller constraints are generated from the UCAs and Loss Scenarios, as demonstrated in Section 3.2.

This final step is designed specifically for aligning the controller constraints extracted directly from the documents and the ones generated from the analysis. In

addition, this step checks if the resulting controller constraints are sufficient to mitigate the identified loss scenarios. Thus, giving the analysts a chance to identify missing or incorrect constraints.

3.3.3 Previous Work

We recently carried out a study to adapt STPA for conformance profiles [66, 67], in which the analysis was performed using a word processing and a diagramming application. Moreover, the hazard analysis was performed in only one part (ordering) of the eReferral workflow, leaving out the reporting part.

Hypothetically, that analysis produced about half of the artifacts that would be generated if the analysis were executed on the entire eReferral workflow. Regardless, even performing the hazard analysis on half of the workflow, the document generated has about 20 pages with tables that extend for many pages and diagrams squeezed to fit a page.

As noted, if the system under analysis is extensive, i.e., includes many functions and interactions, performing the analysis on a word processing application could produce a large document or many documents. Any of these scenarios can be time-consuming to navigate on or search for something.

In addition, STPA is a systematic process in which all artifacts can be traced back and forth among them: Loss Scenarios are caused by UCAs, or Control Actions improperly executed or not executed; UCAs are generated from Control Actions and linked to Hazards; Safety Constraints are linked to Hazards; and, Hazards are linked to Losses. Although word processing software enable links in the documents, maintaining those links during the analysis can be an extra burden for the analysts. Moreover, when the the system under analysis has similar functions, naming can be confusing, and if there is no link among elements, similar names can lead to errors.

In order to mitigate those problems, we believe that the use of specific software to support STPA is highly recommendable and perhaps necessary for large systems. Which leads to our RQ2: *How can we use or adapt hazard analysis tools to support this analysis?*.

Chapter 4

Tool Selection and Adaptation

This chapter supplies a survey of open-source tools intended to assist the application of STAMP techniques and thus help answer the RQ2. These tools are compared in relation to their ongoing development and extensibility. Finally, we present the extensions implemented on FASTEN to assist the application of STPA for the analysis of conformance profiles.

4.1 STPA Tooling Survey

For selecting the tool to perform the hazard analysis we define a set of requirements, which are:

1. The tool should be open-source.
2. The tool should be in active development.
3. The tool should provide the capabilities for extension.

Following the first requirement, we selected the open-source tools XSTAMPP, STAMP Workbench, CAIRIS, WebSTAMP, and FASTEN from our survey. The first three tools are listed on the STAMP website [1], whereas the latter, which are more recent, are still not listed there. It is worth mentioning that the STAMP website contains other tools. However, those are proprietary or closed-source tools, which prevent any extension or adaptation in the tool from people outside the organization that owns it.

[1] `http://psas.scripts.mit.edu/home/stamp-tools/`

4.1.1 XSTAMPP

XSTAMPP (eXtensible STAMP Platform) is an open-source tool created to assist in adopting and using STAMP methodologies in different areas. It was created by a team from the Institute of Software Technology at the University of Stuttgart, Germany, in 2014 [1]. XSTAMPP is written in Java on top of the Eclipse Plug-in-Development Environment (PDE) [2] and Eclipse Rich Client Platform (RCP) [3], making it extensible via plugins. Moreover, XSTAMPP source code is available on GitHub [55]; however, it does not receive any major updates since 2018.

XSTAMPP has seven plugins implemented, including CAST, STPA and STAP-Sec methodologies. These plugins allow users to perform CAST accident analysis, STPA safety analysis and STPA-Sec security analysis. Moreover, an extra plugin, named XSTPA, extends the STPA plugin and adds features to refine and formalize safety requirements into Linear Temporal Logic (LTL) formal specifications to support model verification [2].

XSTAMPP supports STAMP data lists (e.g. hazards, accidents, system goals and design constraints, safety requirements, corresponding safety constraints and control actions), STAMP diagrams (e.g. hierarchical and detailed safety control structures, and process models diagram) and STAMP tables (e.g. unsafe control actions table and causal factors analysis table). Moreover, the main functionalities of XSTAMPP are the following:

- Edit the fundamentals of the analysis.
- Link the components of the analysis.
- Draw the control structure diagram.
- Edit tables for control actions, unsafe control actions and causal factors.
- Augment the control structure diagram with a process model.
- Export and import the STPA hazard analysis result.

4.1.2 STAMP Workbench

STAMP Workbench is an open-source desktop application that supports STPA. It was developed by the Infomation-technology Promotion Agency (IPA), Japan, with the intention of providing an automated and easy-to-use tool accessible for analysts with any level of expertise on STPA [26]. STAMP Workbench is implemented in Java

[2] `https://www.eclipse.org/pde/`

[3] `https://wiki.eclipse.org/Rich_Client_Platform`

on top of the Eclipse Modeling Framework (EMF)[4]. Its source code is distributed as a zip file only, i.e. there is no public repository, and the only way to contact the developers is via email. Moreover, the last major update of STAMP Workbench was in 2021 [26].

STAMP Workbench utilizes tables to define the system preconditions, identify the accidents, hazards and safety constraints, and link the analysis elements. STAMP Workbench automatically generates the control structure diagram through the use of an extraction table. The diagram can also be created using a graphic editor. In addition, STAMP Workbench provides tables for unsafe control actions, hazard causal factors and countermeasures. In summary, the main functionalities of STAMP Workbench are the following:

- Edit the basis of the analysis.
- Guided analysis.
- Link the components of the analysis.
- Generate and draw the control structure diagram.
- Edit tables for unsafe control actions, hazard causal factors and countermeasures.

4.1.3 CAIRIS

CAIRIS (Computer Aided Integration of Requirements and Information Security) is an open-source web platform to elicit, specify, and validate secure and usable systems [8]. It is written in Python and was created by Dr. Shamal Faily from Bournemouth University, England. CAIRIS was not explicitly created or extended to STPA. However, it can be utilized to perform STPA because its concepts are analogous to those required by STPA [9]. CAIRIS source code is available on GitHub [10], and from the commits history, it is receiving updates; however, the latest release was made in December 2020.

CAIRIS supports losses and hazards by defining them as obstacles, and system constraints are defined using goals. To model the control structure, CAIRIS provides a tool for modelling data flow diagrams (DFD). As CAIRIS allows defining custom types for its components, it is possible to adapt them to STPA. In summary, the CAIRIS features that support STPA are the following:

- Edit the components of the analysis.

[4] `https://www.eclipse.org/modeling/emf/`

- Link the components of the analysis.
- Generate the control structure diagram.
- Perform model validation checks.

4.1.4 WebSTAMP

WebSTAMP is a web application that supports STPA and STPA-Sec. It aims to provide a more systematic, automated and comprehensive way to assist safety and security analysis [56]. WebSTAMP is currently being developed collaboratively by many contributors and coordinated by Professor Celso Massaki Hirata from the Instituto Tecnológico de Aeronáutica (ITA), Brazil. WebSTAMP was created using the Laravel[5] web framework built in PHP and other web technologies [21]. Unfortunately, the only way to test the application is via their website and its source code or executable files are not available yet. However, the team's goal is to make WebSTAMP open and available for further extensions [21].

WebSTAMP supports lists for system goals, assumptions, losses, hazards and safety constraints. It provides a tool to model the control structure graphically and text boxes to enter the control structure information, including components and connections. To identify unsafe control actions, WebSTAMP utilizes context tables that verify all possible states of the controlled process. As the context table can grow a lot with the system's complexity, WebSTAMP supports the definition of rules using the process statuses that automatically identify hazardous contexts to fill the table. To identify the loss scenarios, WebSTAMP utilizes a set of customizable guide questions to systematically direct the analysis to find the causal factors and generate recommendations [56]. Moreover, the main functionalities of WebSTAMP are the following:

- Edit the information/purpose of the analysis.
- Link the components of the analysis.
- Allow limited collaborative analysis.
- Define rules to find hazardous context.
- Define guided questions to determine loss scenarios.

[5] `https://laravel.com/`

4.1.5 FASTEN

FASTEN is an open-source environment focused on requirements modelling, formal specification, safety engineering and safety assurance for critical systems [46]. FASTEN was created on top of the Jetbrains Meta Programming System (MPS)[6] language workbench by Daniel Ratiu from Siemens Corporate Technology, Germany, and other contributors. FASTEN is composed of modular and extensible Domain-Specific Languages (DSL). It is designed to be an alternative approach for writing formal specifications at a higher abstraction level than the traditional methods [47]. Besides the DSLs, FASTEN also integrates external analysis tools at the binary level to perform modelling verification [48]. These tools receive the output of the DSLs, abstracting most of the formalism that the modelling verification requires. Moreover, FASTEN source code is available on GitHub [39], and it is in current development, and the latest release is from January 2022.

STPA is one of the safety engineering methods that FASTEN implements. Other examples are Hazard Analysis and Risk Assessment (HARA), Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA). Each of these methods is implemented by various DSLs that can be extended to adapt the method to a specific domain or address a method extension. Moreover, some of the DSLs are shared among methods under the safety engineering package.

For STPA, FASTEN provides lists to define hazards, losses and safety constraints, a graphical editor for drawing the control structure, and a table to identify the unsafe control actions. Furthermore, FASTEN allows linking among STPA components for traceability and verification.

4.1.6 Hazard Analysis Tools Summary

Table 4.1 summarizes the evaluated tools, comparing them using the requirements to pick a tool. Notice that WebSTAMP is not listed in the table because the source code is not available for the general public. From the other four possible candidates, we selected FASTEN because it is the tool with the most active repository, and it has good capabilities of extension via the implementation of Domain-Specific Languages (DSLs).

The tools based on Java/Eclipse would be appropriate options due to their extensibility via Eclipse Plugins. However, STAMP Workbench does not have a public

[6] `https://www.jetbrains.com/mps/`

repository or implementation guide, and despite being recently updated, the major modification was changing the JVM provider [26]. Moreover, the development of XSTAMPP has been mostly stagnant since the latest release [55]. On the other hand, to extend CAIRIS, it is necessary to change the whole system, from database schema to user interface [9].

Tool	**Open Source License**	**Last Release**	**Extensible**	**Platform**
XSTAMPP	EPL-1.0	Nov 2018	Plugins	Java/Eclipse RCP
STAMP Workbench	BSD-2	Dec 2021	Plugins*	Java/Eclipse EMF
CAIRIS	Apache-2.0	Dec 2020	Source Code	Python/Flask
FASTEN	EPL-1.0	Jan 2022	DSLs	Jetbrains MPS

Table 4.1: STPA tools summary.

* Assuming it supports plugins because Eclipse EMF.

4.2 Extending FASTEN

FASTEN is an environment for specification, verification, and assurance of critical systems built on top of Jetbrains MPS. Jetbrains MPS is an open-source integrated development environment (IDE) for language engineering. Jetbrains MPS features a stack of DSLs that can be extended in a modular manner, allowing characteristics like extensibility, reusability, and customization. Therefore, FASTEN inherits these characteristics, making FASTEN an excellent tool to facilitate the execution of hazard analysis using STPA and the other methods it implements.

Although FASTEN implements a set of features that make it ready for STPA, during the analysis, we found that some features could be improved to better display information, and other features did not fulfill the analysis very well; thus, new features were needed. For example, FASTEN implements the loss scenarios as a table containing scenario name, UCA and scenario description. However, we found that presenting the scenarios as a list of requirements is better for clarity. In addition, the scenarios table does not link to control actions; thus, it does not cover all possible loss scenarios defined by FASTEN.

All the extensions implemented for this work are detailed further in this section. Nevertheless, first, we provide an overview of the basic concepts and features of Jetbrains MPS as background for a better understanding of how the extensions work. Notice that this section does not go depth on this topic. Hence neither the MPS base language nor the MPS tooling is detailed here, save when a language component is explicitly utilized or required to understand any implementation. All the Jetbrains MPS environment details can be found in their user guide [7].

4.2.1 Jetbrains MPS Overview

Jetbrains MPS is a toolset for designing domain-specific languages (DSL) based on projectional editing — a term Martin Fowler coined in 2008 [17] — that combines a language workbench and base languages in a modern IDE. On projectional editing, the language's abstract syntax tree is edited directly using projectional representations of the tree.

Abstract Syntax Tree

Abstract Syntax Tree (AST) is a data structure that represents the syntactic structure of the source code written in a programming language. Like other trees in computer science, the AST is composed of nodes, and these nodes have internal properties and links for child nodes. For example, Figure 4.1 shows an AST for the function written in Java to calculate the factorial of an integer recursively, shown in Listing 4.1. Each node in the tree represents a syntactic element of the source code, and each node can have one or many children. For instance, in this example, the *if* statement has a condition and a code statement as children. However, it can have multiple code statements and other children, like *else* and *else if* statements.

[7] `https://www.jetbrains.com/help/mps/mps-user-s-guide.html`

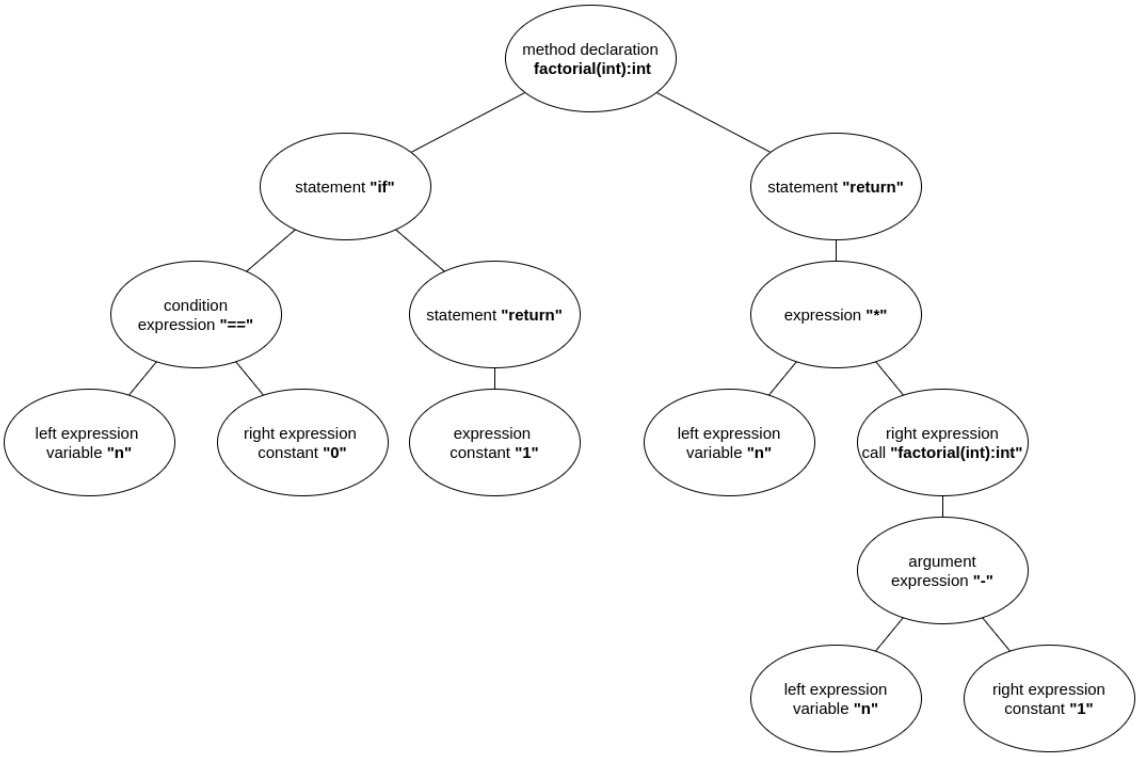

Figure 4.1: AST for a recursive factorial function

Notice that this AST is a simplified tree to illustrate how the source code is represented after being parsed by the interpreter or compiler. Therefore, it does not show many of the details of the nodes and other parts of an actual AST implementation.

```
public static int factorial(int n) {
    if (n == 0) {
        return 1;
    }
    return n * factorial(n - 1);
}
```

Listing 4.1: Recursive factorial function

Whereas General-purpose programming languages utilize parsers to build the AST from the textual source code, MPS manipulates the AST directly from the editor without scanning or parsing the source code.

Projectional Editor

Jetbrains MPS features a projectional editor, which modifies the AST directly when the user changes something on the editor. This projectional editing has many advantages over classical parser-based languages, for example:

1. freedom of notations (we can use in MPS non-parsable notations like tables, diagrams, and other visual components)
2. a concept can have multiple notations
3. languages can be composed without being restricted to grammar ambiguities

Figures 4.2 and 4.3 show projectional editors for a Java Class with a static function to calculate the factorial of a number, written using the MPS base language, and a table of losses written using FASTEN.

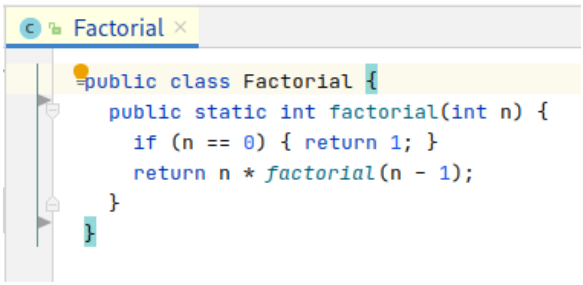

Figure 4.2: Projectional Editor for a factorial function written in Java

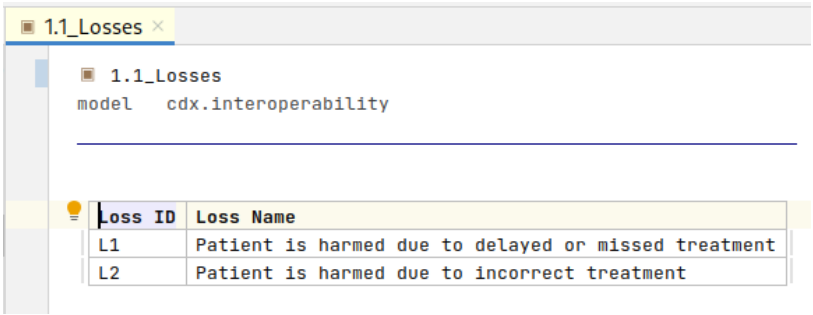

Figure 4.3: Projectional Editor for a table of Losses

As can be noticed, the editor showing the Java code looks exactly like a regular textual representation of a Java Class. However, on a projectional editor, each part of the program is a visual representation of a node of the AST being manipulated. For

instance, Figure 4.4 shows the reflective representation of the *if* statement present in the same factorial function displayed before. This feature of the Jetbrains MPS editor permits the developer to have direct access to the model in a tree-like structure.

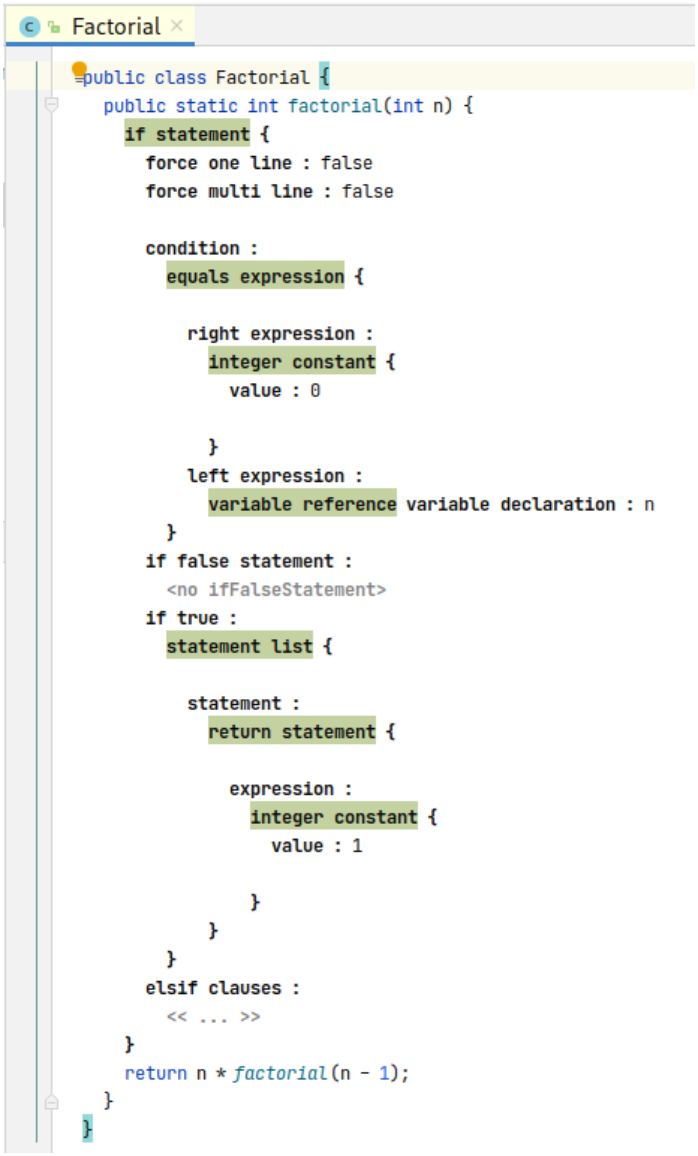

Figure 4.4: Projectional Editor showing the reflective representation of the if statement of a factorial function

Concept and Language aspects

Concepts in MPS are elements that define the class, or type, of the AST nodes and the internal structure of the nodes in that class, including properties, children,

and references. Concepts also define the syntax of a language, building the structure of the language. The **Structure** is the principal aspect of a language built using MPS. It is for MPS what the grammar is for general-purpose languages. The MPS language structure is inspired by object-oriented programming (OOP), and thus it implements some of the OOP principles. For example, a Concept can extend other Concepts or implement *Concept Interfaces*, just like inheritance in OOP; a Concept can be concrete or abstract and has properties (attributes). Furthermore, to build the AST for the language, Concepts can be declared as a *root* node and reference other Concepts as *children*. In addition to the Structure, a language in MPS is composed of other aspects as follows.

The **Editor** defines the characteristics of the projectional editor of a concept, namely, how the concept will be displayed and manipulated. Editors are composed of cells, which can be text, user interface components, or other cells. MPS defines many cells models for designing different concept elements. Moreover, cells are highly customizable through properties that define the cell layout, style, actions and menus. Another editor-related aspect of the language is the **Intention**. This aspect gives the users shortcuts to operations meant to be applied on a specific part of the language, similar to the intention menu of modern IDEs.

The **Constraint** aspect is what its name means. It defines the constraints for a Concept. For example, it defines which instances of a Concept can be child, parent and ancestor of the Constraint's Concept. Constraints can also filter the Concept's properties and define the Concept's scope, i.e. restrict the visibility of the references of a Concept. The **Behavior** aspect is utilized to define instance method, static methods, and constructor for a Concept.

Generators and **TextGen** are MPS aspects that translate language into an output template. Unlike the other aspects of the MPS language, defined as one model by Concept, the Generator aspect can be composed of many models and utilities and can be quite complex, depending on the desired output. Whereas TextGens are simpler models focused on generating text directly from the model language.

MPS Project Structure

Like other IDEs, MPS utilizes the concepts of projects and modules to organize the work. The project is the primary organization unit in MPS, and it holds the modules, which are top-level elements in MPS – the type of a module changes according to

the type of elements it holds (language, solution, devkit, or generator). Modules are composed of many models and hold metadata on the module's properties and dependencies. Model is a lower-level element that holds the code in one or more root nodes.

4.2.2 FASTEN Extensions

As mentioned in Section 4.1, FASTEN is a platform formed by various components, including many DSLs for designing safety systems. This section only shows the FASTEN languages applied to the STPA analysis and thus directly linked to the implemented extensions. Figure 4.5 shows the concept diagram with the FASTEN languages that apply to STPA. It is worth mentioning that this diagram intends to display only the structure of the languages. In addition to these concepts, languages in MPS are built by many other elements.

The `com.mbeddr.formal.safety.hara` package contains concepts utilized to identify Losses and Hazards. The `com.mbeddr.formal.req.base` package contains concepts to create a list of requirements, which can be applied for System-Level Constraints and Controller Constraints. Finally, the `com.mbeddr.formal.safety.stamp` package contains concepts to create Control Structure diagrams and tables of Unsafe Control Actions and Loss Scenarios.

Figure 4.6 shows the concept diagram with the extensions implemented in the FASTEN languages to facilitate the STPA analysis performed in this work. The majority of the implementations are in the `com.mbeddr.formal.safety.stpa` package, save a minor change in the `com.mbeddr.formal.safety.stamp` package. All extensions are explained later on in this section. Moreover, the source code for these extensions artifacts for the SPTA analysis, explained in the next chapter, are available in the GitHub repository `https://github.com/oscarcosta/stpa.icpa`.

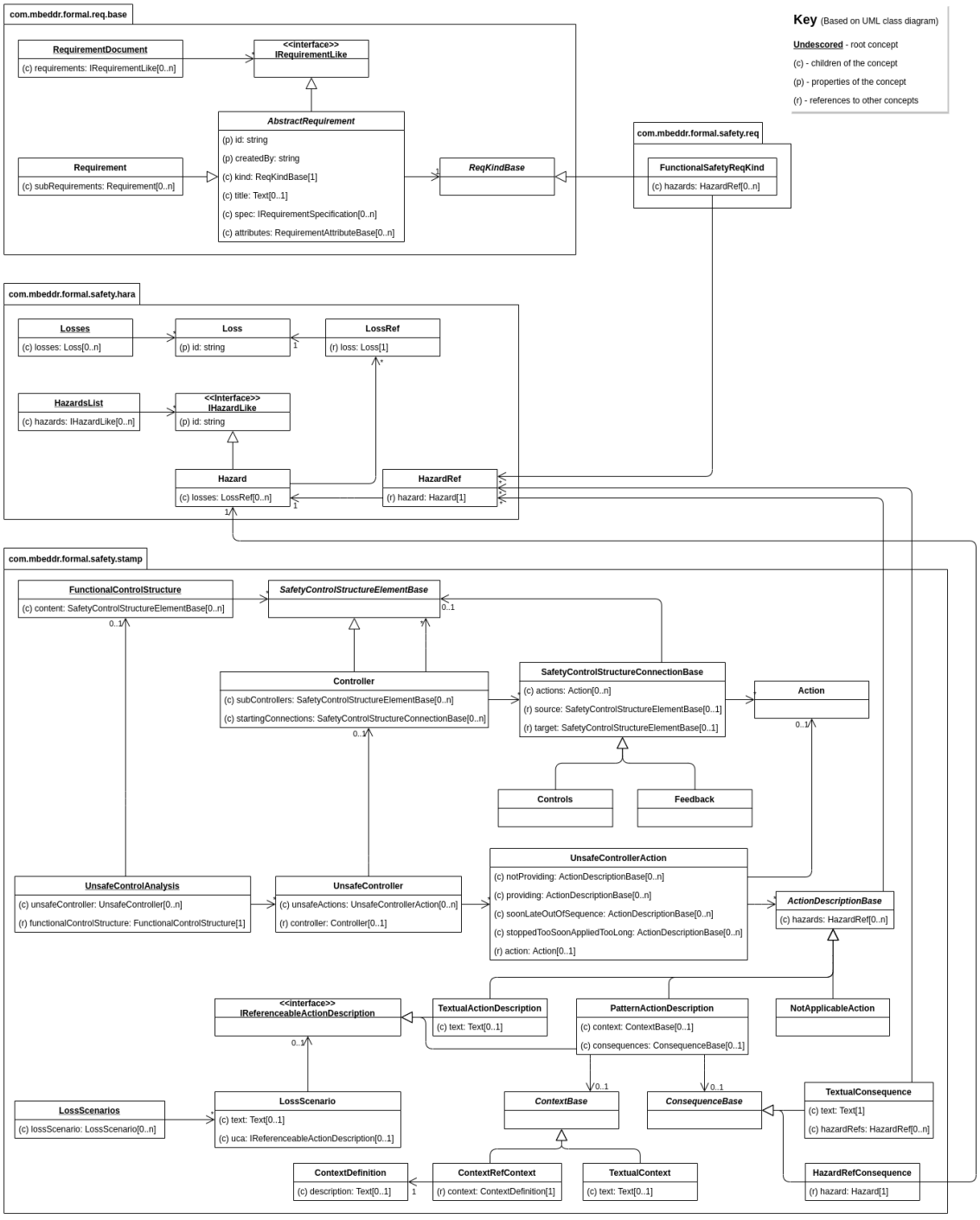

Figure 4.5: FASTEN safety concepts utilized in STPA

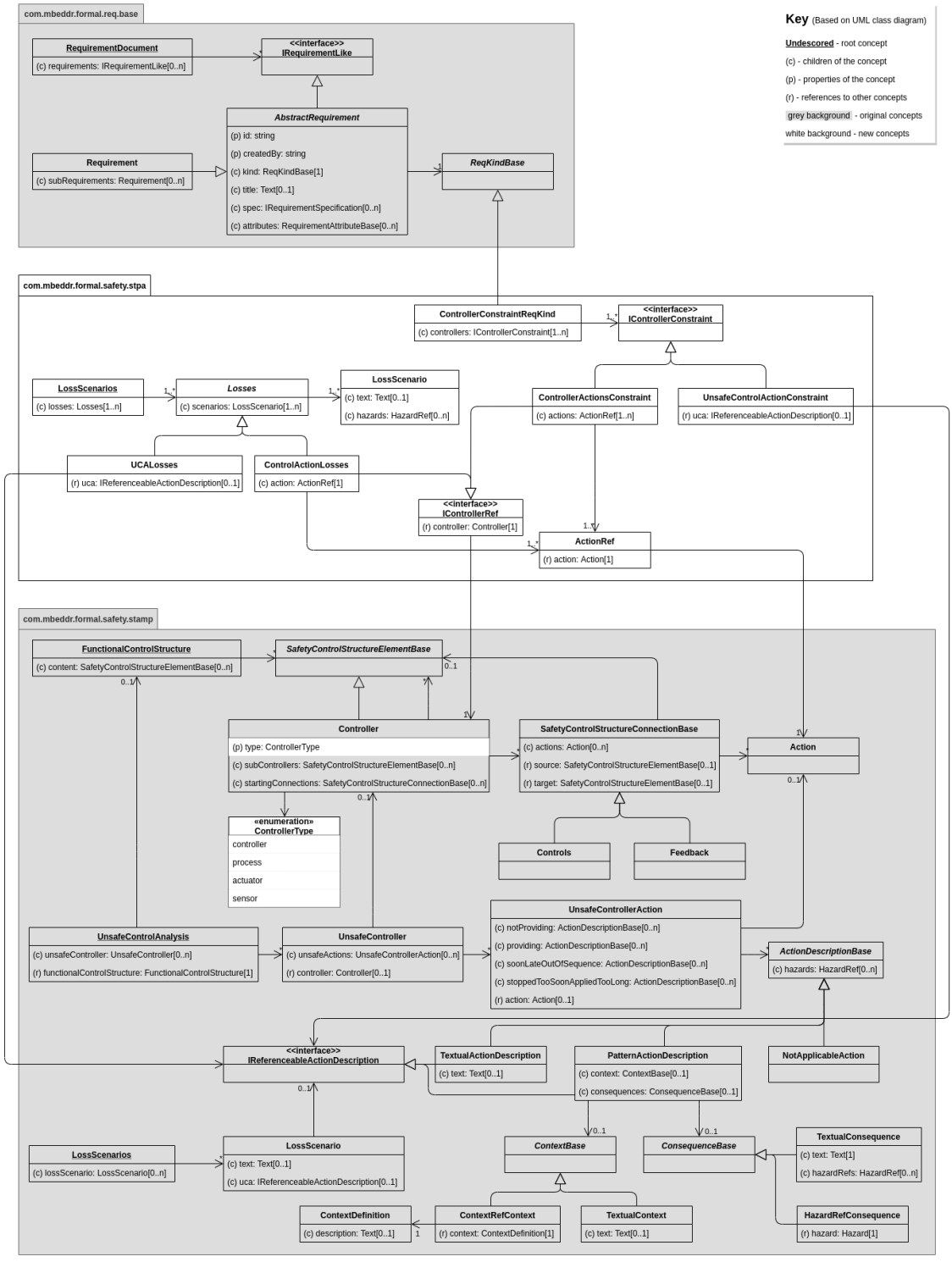

Figure 4.6: FASTEN safety concepts extensions

System-Level and Controller Constraints

FASTEN provided the Requirement concept, which is utilized to define the System Level Constraints and Controller Constraints, among other concepts of other DSLs. This concept defines a child named "kind," which is an abstract concept with multiple implementations. For example, to define the System Level Constraints, the concept utilized is the Functional Safety Requirement., which traces back to the Hazard concept. However, no concept traces back to Control Actions or Unsafe Control Actions, making it difficult to extract the Controller Constraints that link to those elements.

We implemented some concepts that link the constraints to Control Actions or Unsafe Control Actions to define the System Level Constraints and Controller Constraints with the accurate traceability of STPA. The implementation of these concepts is observed in the Figure 4.6, within the package `com.mbeddr.formal.safety.stpa`. The concept *ControllerConstraintReqKind* implements FASTEN's abstract *ReqKindBase* concept, allowing the analyst to choose this type (kind) of requirements when working on the controller constraints. Furthermore, the *ControllerConstraintReqKind* can be associated with Control Actions and UCAs via the concepts *ControllerActionsConstraint* and *UnsafeControlActionConstraint*. Figure 4.7 shows examples of the utilization of these concepts.

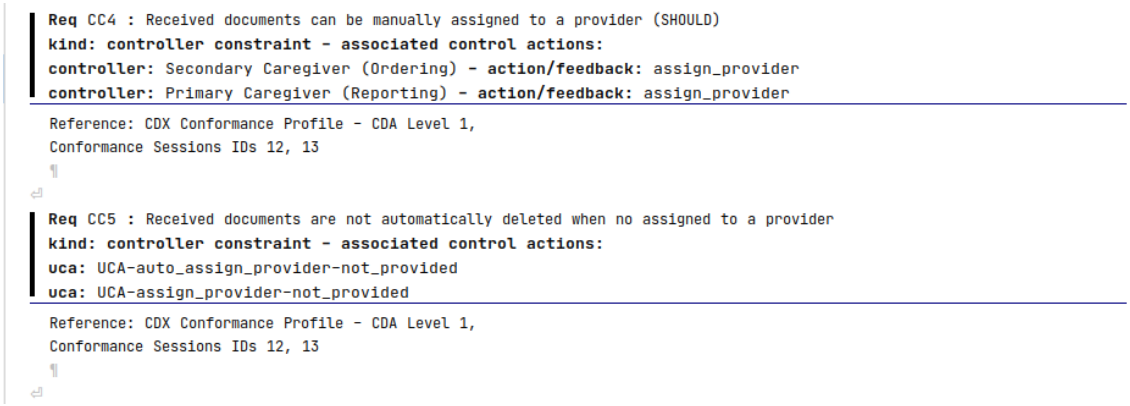

Figure 4.7: Controller Constraint examples

FASTEN provides a feature to serialize requirement documents into Microsoft Word format. However, this feature only exports the *title* and *specs* children of the Requirement model, keeping out the other model children, namely *kind* and *attributes*, which holds important information for the system-level constraints and controller constraints artifacts of the STPA analysis.

To solve that problem, we implemented a set of generators in the extension language using the `com.dslfoundry.plaintextgen` [8] MPS plugin to export all requirement's properties. This plugin allows exporting the requirements document in the more flexible Markdown [9] format. The Markdown format was selected because it can be easily converted to any other text format. For example, the Appendices of this thesis were exported from MPS and converted from Markdown to LaTeXusing MultiMarkdown [10].

Control Structure

The FASTEN control structure simplifies the STPA control structure by defining only concepts for Controllers, Control Actions and Feedbacks. This simplification is acceptable for simple models like software systems where Actuators and Sensors are not utilized or can be abstracted. However, the lack of the Controlled Process concept makes it difficult to identify the Unsafe Control Actions and Loss Scenarios, which depend on this concept. For that reason, we created an enumeration named *ControllerType*, which defines the possible types of components, in order to improve the FASTEN control structure. Then, this enumeration was included as a property named *type* in the *Controller*, as shown in Figure 4.6 in the package `com.mbeddr.formal.safety.stamp`. Furthermore, the concept editor of the Controller concept was modified to allow the user to edit and visualize the type in the diagram. An example of the control structure diagram modified is shown in Figure 4.8.

[8] `https://plugins.jetbrains.com/plugin/8444-com-dslfoundry-plaintextgen`
[9] `https://daringfireball.net/projects/markdown/`
[10] `https://fletcherpenney.net/multimarkdown/`

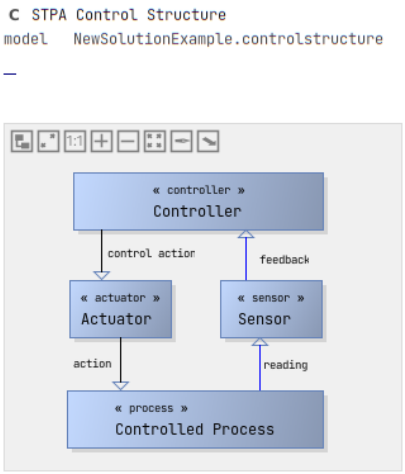

Figure 4.8: Control Structure example

Loss Scenarios

FASTEN implements the STPA Loss Scenarios as a table composed of scenario name, UCA and scenario description, and the scenario description can have links to requirements (constraints) only. This implementation may lead the analyst to identify only scenarios directly connected to UCAs, limiting the identification of Loss Scenarios related to Control Actions, but not to UCAs. Furthermore, due to the lack of links to other elements, such as Controllers, Actions, and Hazards, traceability among the STPA artifacts is compromised.

Due to that, we implemented a new structure of concepts for handling all types of loss scenarios described in the STPA Handbook [34]. These concepts are shown in Figure 4.6 in the package `com.mbeddr.formal.safety.stpa`. The root concept LossScenarios has a list of losses, a list of the abstract concept Losses. Then, Losses is extended by the concepts UCALosses and ControlActionLosses. These concepts respectively implement the STPA Loss Scenarios that lead to UCAs and the STPA Loss Scenarios related to a Control Action and not linked to any UCA, as shown in Figure 4.9.

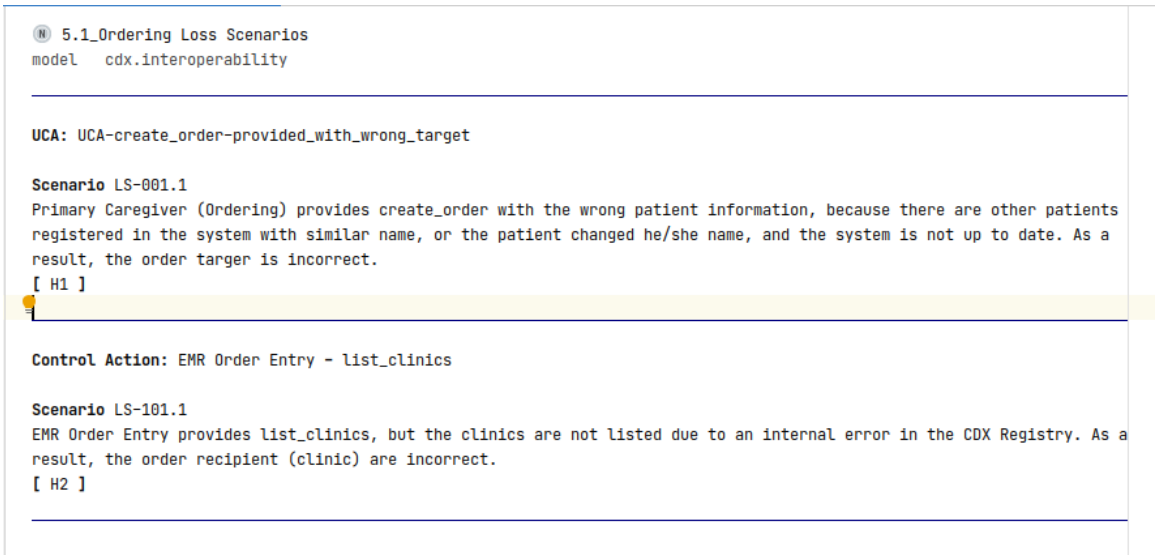

Figure 4.9: Loss Scenarios example

The new language structure for loss scenarios includes a checking rule for verifying the uniqueness of the loss scenario's name, similar to checking present in the UCAs table in FASTEN. In addition, the new language structure has an *intention* to copy the text and hazards from the UCA related to the loss scenarios. This intention facilitates the writing of loss scenarios that have similar writing, or use the UCA content:

Loss Scenario = <UCA>, <Scenario> [Link to Hazards]

Additional Text Generators

In addition to the generators implemented for exporting the requirement documents described before, we implemented two additional sets of generators to export the tables of UCAs and the new Loss Scenario documents. These generators also use the `com.dslfoundry.plaintextgen` plugin and export the documents in the Markdown format.

Chapter 5

Applying STPA to CDX eReferral Workflow

As previously mentioned, the CDX system's primary goal is to facilitate the exchange of clinical documents among health providers systems across the province. This electronic data exchange replaces paper-based communication, improving the data exchange process and, more importantly, bringing benefits from the clinical and patient care perspective. These benefits include the timely delivery of medical results and information to assist and speed up diagnoses, and the improvement of patients' medical history, which helps with patient care and treatment plans [36].

Notice that the CDX interoperability operates with very personal and sensitive information, and any unexpected modification or loss of patient health information can jeopardize the patient's health if they do not receive adequate treatment as a result of tampered or lost information. Also, serious problems can occur with the patient or clinics if bad actors get their hands on the patient's private information. Therefore, the EMR systems that implement the CDX interoperability are tested and certified through conformance profiles that check various aspects of the implementation, such as compatibility with the standards and system safety and security.

Conformance profiles are formulated straight from the system specifications. However, those conformance profiles are usually restrained to a sub-set of the specification due to scope restriction [42]. This scope restriction could compromise the system safety because safety-related requirements were unintentionally left out from the conformance profiles. Furthermore, if any safety-related constraint was not foreseen in the original system requirements, the conformance profiles will also lack these con-

straints, hence jeopardizing the safety of the system.

Besides being utilized as testing guidelines, conformance profiles are also a source of system requirements by system vendors to implement interoperability in their systems [42]. Therefore, these requirements are crucial for the quality and safety of the systems. These safety concerns raise the question of how to ensure conformance profiles correctness and completeness.

Therefore, we adapted the STPA method for validating the CDX interoperability conformance profiles [66, 67]. However, that previous work was done manually and only considered the ordering part of an eReferral workflow, and thus raised some concerns about the manual approach for performing hazard analysis, as mentioned in Chapter 3. Therefore, this work aims to address those concerns by expanding the case study and applying the STPA adaptation (described in Section 3.3) with the specialized tooling FASTEN (described in Section 4.2) to the CDX System in order to evaluate the safety of the CDX interoperability conformance profiles. In addition, in this study, the hazard analysis contemplates the two parts of the eReferral workflow.

5.1 System Description

The CDX system supports several clinical workflows through the exchange of CDA messages [11]. Those clinical workflows can be the simple exchange of clinical notes, the submission of test results or a more complex workflow, such as a clinical referral. In this work, we chose to apply the extension of the original STPA described in Section 3.2 on the eReferral workflow implemented using the CDX interoperability system.

The eReferral workflow is the electronic version of the clinical referral that utilizes EMR systems to transmit messages between health care providers. Overall a referral workflow starts after a first consultation with the patient's primary care provider. After seeing the patient, the primary care provider sends an order containing the patient's information and health condition in a referral to a secondary provider whose specialization is in the patient's health issue. Then, this specialist sees the patient for a consultation or treatment and reports back the diagnosis to the first practitioner. Moreover, both practitioners can also exchange additional messages, notes, and exam results during the treatment.

5.2 Defining the Purpose of the Analysis

The purpose of the analysis consists of identifying the Losses, System-Level Hazards and System-Level Constraints for the system under investigation. For this industry case, the system boundaries are defined by the eReferral workflow mentioned in the previous section. Moreover, this analysis uses the system specification documents, CDX Technical Overview [11] and CDX Conformance profile [50], as input.

Both the workflow and the documentation help abstract the system for this analysis. For instance, the system abstraction utilized in this analysis does not consider implementation details such as the application platform (web, desktop, mobile or cross-platform), how the user interface is implemented, or how the data is stored.

For this step, FASTEN provides tables for recording the losses and system-level hazards and a list for the system-level constraints. This latter is a list of requirements of type "functional safety," which lets us link the constraint to one or more hazards. In addition, the table of hazards has a column to connect the hazard to one or more losses.

5.2.1 Losses

In the context of the eReferral workflow, the identified losses are related to the patient being harmed by an incorrect medical intervention, delayed or not happening due to a system failure or an unsafe interaction of system components. The losses are shown in Figure 5.1. Nonetheless, it should be emphasized that those are not the only possible losses for this system. As discussed in Section 3.2, the elected losses help define the purpose of the analysis. Therefore, for this analysis, the patient's health is the only aspect considered.

Loss ID	Loss Name
L1	Patient is harmed due to delayed or missed treatment
L2	Patient is harmed due to incorrect treatment

Figure 5.1: Losses (all)

5.2.2 System-Level Hazards

Figure 5.2 shows the system-level hazards identified by the analysis. As can be noticed, all hazards, but the last one, address both order and report in the same

manner. This decision is possible because the CDX system handles any message in the same way. Moreover, only a few characteristics differentiate orders from reports in the EMR systems involved in the communication, as we will observe in the next steps of the analysis.

Hazard ID	Hazard Name	Associated losses
H-01	Order/report target (patient) is incorrect or incomplete	L1, L2
H-02	Order/report recipient (clinic or provider) is incorrect	L1
H-03	Order/report content (body or attachment) is incorrect or incomplete	L1, L2
H-04	Order/report is delayed, not delivered, or lost	L1
H-05	Order/report is duplicated	L2
H-06	Report is not linked to the correct order	L1, L2

Figure 5.2: System-Level Hazards (all)

5.2.3 System-Level Constraints

The system-level constraints defined in this step aim to address the system-level hazards previously identified. Thus, each constraint is somehow the inverse meaning of each hazard. Still, it is not necessary to define one constraint for each hazard. I.e. one constraint can address one or more hazards. Figure 5.3 shows the system-level constraints defined to prevent the identified hazards.

Req SC-01 : Order/report target (patient information) is correct and complete
kind: functional safety - addressed hazards H-01

Req SC-02 : Order/report recipient (provider or clinic) is correct and valid
kind: functional safety - addressed hazards H-02

Req SC-03 : Order/report is valid and consistent
kind: functional safety - addressed hazards H-03

Req SC-04 : Order/report content is correct and accurate
kind: functional safety - addressed hazards H-03

Req SC-05 : Order/report is timely delivered
kind: functional safety - addressed hazards H-04

Req SC-06 : Order/report duplication is prevented
kind: functional safety - addressed hazards H-05

Req SC-07 : Report is linked to correct order
kind: functional safety - addressed hazards H-06

Figure 5.3: System-Level Constraints (all)

5.3 Modelling the Control Structures

In step 2 of STPA, we model the system under analysis in a control structure formed by control loops. The FASTEN control structure represents the controllers, controlled processes, actuators and sensors as boxes. The control actions are black arrows, and the blue arrows are the feedbacks.

Figure 5.4 show the high-level control structure of a referral workflow. This control structure is composed of primary and secondary caregivers (controllers), the patient health (controlled process), ordering (actuator) and reporting (sensor). Furthermore, the primary caregiver also "controls" the second caregiver through the ordering actuator and receives feedback via the reporting sensor.

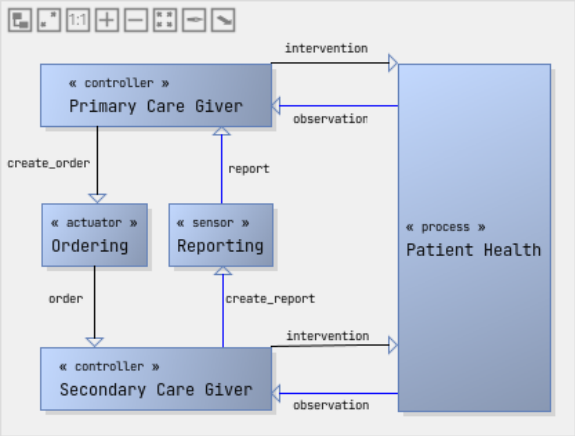

Figure 5.4: Referral Control Structure

This high-level control structure can be applied to any referral workflow. Thus, we decompose this control structure to fit the analyzed eReferral workflow better.

For the eReferral workflow, the ordering and reporting parts of the control structure are performed by EMR systems, which are integrated somehow. As described before, the CDX system facilitates the eReferral workflow by providing a safe and secure way for two or more EMR systems to interoperate. Hence, both ordering and reporting are composed of EMR systems and the CDX system.

Therefore, we broke down the control structure to expose the systems responsible for the ordering and reporting, as ilustrated in Figures 5.5 and 5.6, respectively. This task is performed using as input the CDX Technical Overview [11], which describes the inbound and outbound interfaces and the logical behaviour of the CDX System.

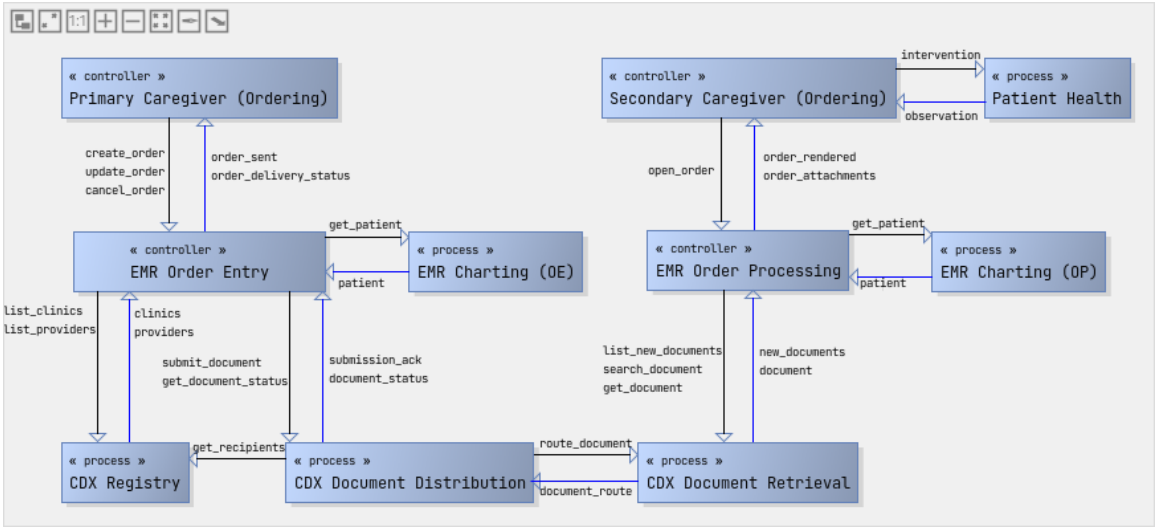

Figure 5.5: Ordering Control Structure

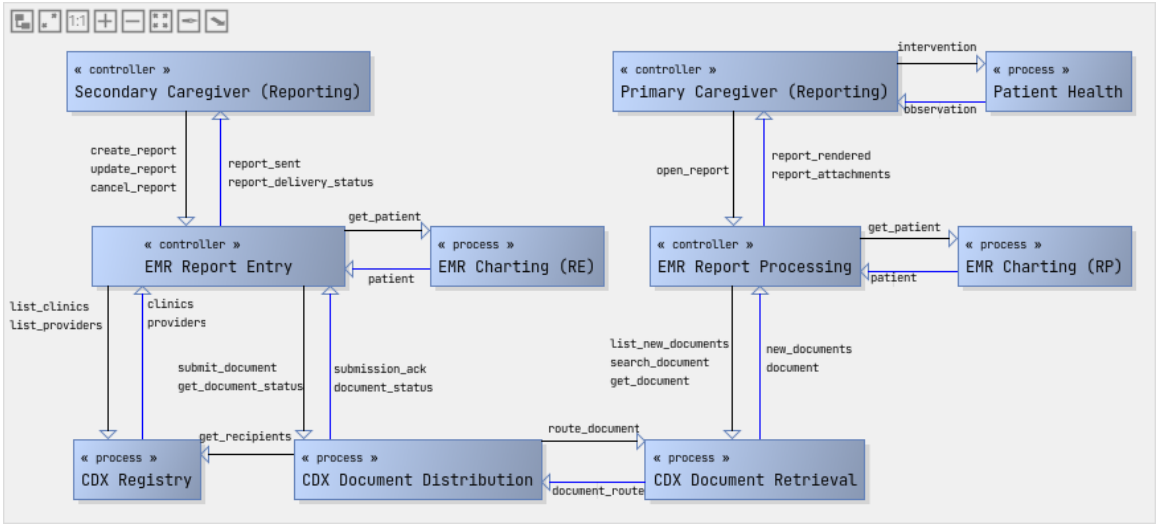

Figure 5.6: Reporting Control Structure

Note that the two control structures are nearly identical, only changing who sends and receives orders or reports and the type of document exchanged. This happens because the CDX system does not differentiate ordering from reporting. Using both control structures for the remainder of the analysis is possible, but it would duplicate many of the artifacts of the following steps. Due to that, we represent both components using only one control structure, shown in 5.7, that handles orders and reports as documents.

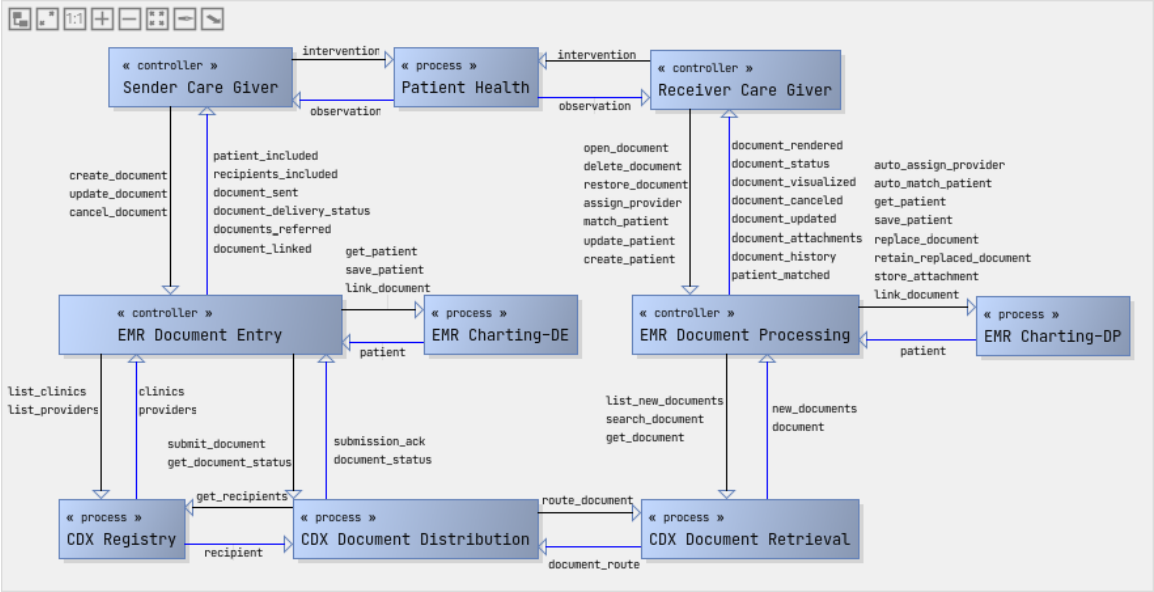

Figure 5.7: Document Exchange Control Structure

Notice that the Primary and Secondary Care Givers are not present in this diagram. Instead, this control structure shows a Sender Care Giver and a Receiver Care Giver, respectively, responsible for sending and receiving documents submitted through the CDX system. Moreover, it is worth mentioning that the roles of sender and receiver are defined in the CDX Conformance Profile — the guidance document for the certification of the CDX integration — as well as the new control actions and feedbacks included in the latter control structure. Therefore, this control structure is modelled according to what is expected by the conformance process.

A most significant difference between the control structures is that in the eReferral control structure, control actions have a circular flow — where the order passes from the Primary Care Giver to the Secondary Caregiver, and the report does the contrary path. In contrast, in the CDX document exchange control structure, documents are sent asynchronously. I.e. the Sender Care Giver sends a document to the CDX systems. Then, the Receiver Caregiver actively retrieves that document from the CDX system.

In addition to the human controllers and processes, this controller structure shows components of both EMR systems, sender and receiver, and a few components representing the CDX system. For each of the EMR systems, there is a controller and a charting process. The controllers represent logical parts of the EMR systems responsible for the entry and processing of electronic documents. The charting processes

are logical components where the patient information is manipulated. Furthermore, the CDX system is represented by three controlled processes: CDX Registry, CDX Document Distribution and CDX Document Retrieval. Nonetheless, note that the CDX system could be described using only one process. However, this separation was an assumption made following the CDX web services description.

5.4 Identifying Controller Constraints

Different from the original STPA method, which usually identifies the controller constraints based on the unsafe control actions, in this step, we use the system documentation and the control structure to identify the controller constraints.

At this point of the analysis, we gathered the constraints from the CDX Conformance Profile, which describes 62 conformance requirements utilized to certify the compliance of the EMRs with the CDX system. Those requirements are written using SHALL to indicate mandatory requirements, SHOULD for recommended requirements, and MAY for optional requirements. Moreover, the requirements cover various aspects of the certification, such as CDA document syntax, security, document handling, document ontology and standards. However, due to the scope of this analysis, we only consider the safety-related requirements. In addition, we identified some safety-related constraints from the CDX Technical Overview that were missing in the CDX Conformance Profile.

FASTEN stores the controller constraints as a list of requirements of the type "controller constraint," which allows us to link the constraint to control actions or UCAs. In addition, the requirements have an area to input textual specifications, where we utilized for annotations regarding the documents, updates and missing constraints. Figure 5.8 shows an excerpt of the controller constraints identified in this step. The whole list of constraints is in the Appendix A. Notice that the list is in its final state, with some requirements already updated.

44 controller constraints were extracted from 50 safety-related conformance criteria from the CDX Conformance Profile. As can be observed, it is not a one-to-one relation because some constraints address multiple criteria, and some criteria generate more than one constraint.

Req CC-001 : Standardized documents are received.
kind: controller constraint - associated control actions:
controller: EMR Document Processing - **action/feedback:** get_document
controller: CDX Document Retrieval - **action/feedback:** document

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Session IDs 1, 2, 3, 4, 38

Req CC-002 : Standardized documents are rendered.
kind: controller constraint - associated control actions:
controller: Receiver Care Giver - **action/feedback:** open_document
controller: EMR Document Processing - **action/feedback:** document_rendered

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 1, 2, 3, 4, 28, 38

Req CC-003 : Received documents are automatically assigned to at least one provider.
kind: controller constraint - associated control actions:
controller: EMR Document Processing - **action/feedback:** auto_assign_provider

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 12, 13

Req CC-004 : Received documents can be manually assigned to a provider (SHOULD).
kind: controller constraint - associated control actions:
controller: Receiver Care Giver - **action/feedback:** assign_provider

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 12, 13

Req CC-005 : Received documents are not automatically deleted when no assigned to a provider.
kind: controller constraint - associated control actions:
controller: Receiver Care Giver - **action/feedback:** assign_provider
controller: EMR Document Processing - **action/feedback:** auto_assign_provider

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 12, 13

Req CC-006 : Received documents are automatically matched to an existing patient using the 4 point matching criteria.
kind: controller constraint - associated control actions:
controller: EMR Document Processing - **action/feedback:** get_patient, patient_matched

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 14

Figure 5.8: Controller Constraints (excerpt)

In addition, The constraints *"CC-016"* and *"CC-040"* shown in Figure 5.9, are defined in the CDX Conformance Profile as optional (MAY). However, these constraints should be at least recommended (SHOULD) since the non-cancellation of clinical documents that should be cancelled can lead to hazardous situations.

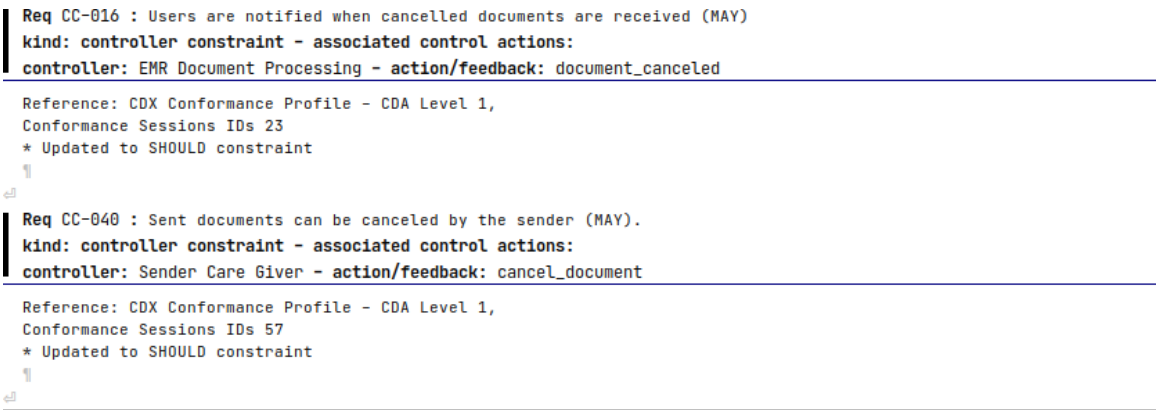

Figure 5.9: Updated Controller Constraints (excerpt)

We also gathered 13 constraints from the CDX Technical Overview document. However, the text of this document is not written in the form of requirements such as the Compliance Profile. Thus, we extract the constraints using a qualitative approach, in which we categorize the text excerpts that better fit the topics that are interesting for the analysis. Then, we create constraints from these excerpts. For instance, the constraint *"CC-054: Receiving system is responsible for keeping track of the listing of new documents."* and *"CC-055: Receiving system is responsible for download and store documents from CDX."* should be included in the CDX Conformance Profile. These constraints ensure that the system automates the receiving and storing of CDX documents. Keeping the user in charge of this action could lead to hazards, such as delayed orders or reports.

The CDX Conformance Profile has a number of constraints directed to the attachments. However, it does not mention anything about the cryptographic hash of the attachment. Hash functions are usual ways of verifying the integrity of files. Moreover, the CDA XML scheme already expects the hash of the attachment. Therefore, the constraint *"CC-025: Attachments being sent are secured by a cryptographic hash function."* should be included in the CDX Conformance Profile. Figure 5.10 shows these three missing controller constraints.

```
Req CC-025 : Attachments being send are secured by cryptographic hash function.
kind: controller constraint - associated control actions:
controller: EMR Document Entry - action/feedback: submit_document

  * REference: CDA XML Schema.
  * Missing constraint in conformance profile!
  ¶
⏎
Req CC-054 : Receiving system is responsible for keeping track of the listing of new documents.
kind: controller constraint - associated control actions:
controller: EMR Document Processing - action/feedback: list_new_documents
controller: CDX Document Retrieval - action/feedback: new_documents

  Reference: CDX Technical Overview,
  Section 3.5.2.1
  * Missing constraint in conformance profile!
  ¶
⏎
Req CC-055 : Receiving system is responsible for download and store documents from CDX.
kind: controller constraint - associated control actions:
controller: EMR Document Processing - action/feedback: get_document
controller: CDX Document Retrieval - action/feedback: document

  Reference: CDX Technical Overview,
  Section 3.5.2.1
  * Missing constraint in conformance profile!
  ¶
⏎
```

Figure 5.10: Missing Controller Constraints (excerpt)

5.5 Identifying Unsafe Control Actions

The identification of unsafe control actions (UCAs) is performed with the help of the control structure, as described in Section 3.2. In this step, FASTEN helps by linking a control structure diagram in the UCA document. Then, it verifies if all control actions of an included controller are present in the UCAs table, warning the user otherwise.

Figure 5.11 shows an excerpt of the UCAs table for the CDX document exchange control structure. The whole table is in the Appendix B as a textual table because an image of the table would be too big to fit on a page. Notice that the FASTEN's UCAs table shows the *Source Controller*, its *actions*, and the four types of UCA defined by STPA — *"Not Providing Causes Hazard"*, *"Providing Causes Hazard"*, *"Too Soon/Late, Out of Sequence"*, and *"Stopped to Soon, Applied too Long"*. Moreover, the UCAs contain an identifying name, the constraint text and links for the addressed hazards. Finally, the *@controller* and *@action* within the text are optional links for controllers and actions, respectively.

Unsafe Control Analysis for Control Structure: 2.2_Document Exchange Control Structure

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazard	Too Soon/Late, Out of Sequence	Stopped too Soon, Applied too Long
Sender Care Giver	create_document	N/A	**UCA-create_document-provided_with_wrong_target** **@controller**(Sender Care Giver) provides **@action**(create_document) with the wrong target information [H-01] **UCA-create_document-provided_with_wrong_recipient** **@controller**(Sender Care Giver) provides **@action**(create_document) with the wrong recipient (provider or clinic) [H-02, H-04] **UCA-create_document-provided_with_wrong_content** **@controller**(Sender Care Giver) provides **@action**(create_document) with the wrong content [H-03] **UCA-create_document-provided_for_wrong_order** **@controller**(Sender Care Giver) provides **@action**(create_document) (report) for the wrong document (order) [H-06]	N/A	**UCA-create_document-provided_repeatedly** **@controller**(Sender Care Giver) provides repeatedly **@action**(create_document) for the same document [H-05] **UCA-create_document-stopped_too_soon** **@controller**(Sender Care Giver) stop providing **@action**(create_document) before **@controller**(EMR Document Entry) conclude **@action**(submit_document) [H-01, H-02, H-03, H-04]
	update_document	N/A	**UCA-update_document-provided_for_wrong_document** **@controller**(Sender Care Giver) provides **@action**(update_document) for the wrong document [H-04] **UCA-update_document-provided_with_wrong_target** **@controller**(Sender Care Giver) provides **@action**(update_document) with the wrong target [H-01] **UCA-update_document-provided_with_wrong_recipient** **@controller**(Sender Care Giver) provides **@action**(update_document) with the wrong recipient (provider or clinic) [H-02, H-04] **UCA-update_document-provided_with_wrong_content** **@controller**(Sender Care Giver) provides **@action**(update_document) with the wrong content [H-03]	N/A	**UCA-update_document-provided_repeatedly** **@controller**(Sender Care Giver) provides repeatedly **@action**(update_document) for the same document [H-05] **UCA-update_document-stopped_too_soon** **@controller**(Sender Care Giver) stop providing **@action**(update_document) before **@controller**(EMR Document Entry) conclude **@action**(submit_document) [H-03, H-04]
	cancel_document	N/A	**UCA-cancel_document-provided_for_wrong_document** **@controller**(Sender Care Giver) provides **@action**(cancel_document) for the wrong document [H-04]	N/A	**UCA-cancel_document_stopped_too_soon** **@controller**(Sender Care Giver) stop providing **@action**(cancel_document) before **@controller**(EMR Document Entry) conclude **@action**(submit_document) [H-04]
	intervention	N/A	N/A	N/A	N/A

Figure 5.11: UCAs (excerpt)

In total, we identified 54 UCAs from 5 controllers (Sender Care Giver, EMR Document Entry, CDX Document Distribution, Receiver Care Giver, and EMR Document Processing). It is worth mentioning that EMR Chartings, CDX Registry and CDX Document Retrieval do not expose UCAs because these are controlled processes and do not have control actions.

5.6 Identifying Loss Scenarios

After having identified the UCAs, we work in the identification of the loss scenarios, using the UCAs and the control structure as aid. Since the control structure under analysis, shown in Figure 5.7, has two paths of control actions, we decided to group the loss scenarios in two documents. The first document contains the loss scenarios related to the document submission, and the scenarios from the document retrieval comprise the other document. This separation is not needed, but it helps handle the documents because the number of scenarios can grow considerably in relation to the number of UCAs.

As described in Section 4.2, we implemented a set of concepts in a new FASTEN

DSL to handle the loss scenarios. This new structure defines lists of scenarios, which can be linked to a UCA or control action.

Many of the loss scenarios are already mitigated by controller constraints or addressed by different scenarios. Thus, this section only describes the loss scenarios that require new constraints to be avoided. Still, the entire content of both documents is in the Appendix C.

In total, 44 loss scenarios require additional constraints to be avoided. Figure 5.12 shows loss scenarios from the document submission that required new constraints, and Figure 5.13 shows scenarios in the same situation but from document retrieval.

```
UCA: UCA-create_document-stopped_too_soon

Scenario LS-SD-006.1
 @controller(Sender Care Giver)  stop providing  @action(create_document) because they perform some action that aborts
the document creation believing that they are concluding the document creation.
* Mitigated by @req(CC-032)
* New @req(CC-105)
[ H-01, H-02, H-03, H-04 ]

Scenario LS-SD-006.2
 @controller(Sender Care Giver) stop providing  @action(create_document) because they attempt to conclude creating the
document, but the document is not submitted due to some connection error, and they are not informed or do not see the
error message and close the form.
* Mitigated by @req(CC-032)
* New @req(CC-106)
[ H-01, H-02, H-03, H-04 ]
```

```
UCA: UCA-update_document-provided_for_wrong_document

Scenario LS-SD-007.1
 @controller(Sender Care Giver)  provides  @action(update_document)  for the wrong document because they pick the wrong
document from the correct patient or a patient with a similar name.
* Mitigated by @req(CC-028), @req(CC-030), @req(CC-037) and @req(CC-038)
* New @req(CC-108)
[ H-04 ]

Scenario LS-SD-007.2
 @controller(Sender Care Giver) provides  @action(update_document) for an outdated version of the correct document,
losing the information contained in the newest versions.
* New @req(CC-109)
[ H-03 ]
```

Figure 5.12: Loss Scenarios for document submission (except)

```
UCA: UCA-list_new_documents-not_provided

Scenario LS-RD-012.1
 @controller(EMR Document Processing)  does not provide  @action(list_new_documents)  because the EMR System is not
configured to list new documents automatically.
* Mitigated by @req(CC-054)
* New @req(CC-112)
[ H-04 ]
```

```
UCA: UCA-list_new_documents-provided_too_late

Scenario LS-RD-013.1
 @controller(EMR Document Processing)  provides too late  @action(list_new_documents)  because the time interval is not
short enought or the EMR system is not configured to list new documents automatically.
* Mitigated by @req(CC-054)
* New @req(CC-112)
[ H-04 ]
```

```
UCA: UCA-search_document-not_provided

Scenario LS-RD-014.1
 @controller(EMR Document Processing)  does not provide  @action(search_document)  for a missed document (due to a fail
in @action(get_document) ) because the EMR System does not keep track of documents that could not be retrieved.
* New @req(CC-115)
[ H-04 ]
```

```
UCA: UCA-get_document-not_provided

Scenario LS-RD-015.1
 @controller(EMR Document Processing)  does not provide  @action(get_document)  for a new document because the EMR
system is not configured to get documents automatically.
* Mitigated by @req(CC-055)
* New @req(CC-113)
[ H-04 ]
```

Figure 5.13: Loss Scenarios for document retrieval (excerpt)

The red underline on the links for controllers and actions in the loss scenarios is a bug/limitation in the tooling. The same behaviour also happens for the FASTEN loss scenarios table. The correct fix would require some modifications on the FASTEN DSLs. We chose not to implement the fix now because the affected links are optional elements in the loss scenario, and the loss scenario concept is not used for any verification other than the scenario id uniqueness check. However, it would be interesting to fix it if the new DSL is integrated into the FASTEN codebase.

We also identified 11 loss scenarios that, if they happen, there is no possible mitigation besides trying to execute the action once again or at another time. Figure 5.14 shows some of these scenarios. Notice that some of these scenarios are not associated with the UCAs but linked to a Control Action. This condition is expected

by STPA, and it happens when the controller provides the action, but the action is improperly executed or non-executed by the controlled process.

UCA: UCA-submit_document-stopped_too_soon

Scenario LS-SD-017.1
@controller(EMR Document Entry) stops too soon providing **@action(submit_document)** before completing the document submission because there is a communication problem with the CDX system.
* Suggestion: System or users should decide if they will try it again when the connection is re-established.
[H-04]

Control Action: EMR Document Entry - list_providers

Scenario LS-SD-030.1
@controller(EMR Document Entry) provides **@action(list_providers)** , but the clinics are not listed due to a connection issue or error on the CDX system.
* Mitigated by **@req(CC-058)**
* Suggestion: System or users should decide if they will try it again when the connection is re-established.
[H-02]

Control Action: EMR Document Entry - submit_document

Scenario LS-SD-031.1
@controller(EMR Document Entry) provides **@action(submit_document)** , but the submission acknowledgment is not returned due to a connection issue or error on the CDX system.
* Mitigated by **@req(CC-058)**
* Suggestion: Resubmit a document without being sure that the previous attempt was unsuccessful will cause the document to be duplicated at the recipient side.
[H-04]

Control Action: EMR Document Entry - get_document_status

Scenario LS-SD-032.1
@controller(EMR Document Entry) provides **@action(get_document_status)** , but the document status is not returned due to a connection issue or error on the CDX system.
* Mitigated by **@req(CC-058)**
* Suggestion: System or users should decide if they will try it again when the connection is re-established.
[H-04]

Figure 5.14: Loss Scenarios for document submission not solved (excerpt)

These scenarios could happen in the presence of any problem that prevents the actions from being executed correctly in the CDX system or the correct feedback from reaching the EMR system. Thus, knowing that those control actions are network calls and assuming that the list_clinics, list_providers, get_document_status, list_new_documents and search_documents actions do not modify any information on the server-side (CDX system), they could be classified as idempotent operations. Therefore, a first attempt to mitigate any connection-related issue would be to resubmit the original message in a reasonable amount until the system gets the ex-

pected response. However, it is known that the control actions submit_document and get_document modify server-side information. Thus, to use the same solution, those operations should be changed to be idempotent operations in the CDX system, or a more complex mitigation scenario should be implemented in the EMR system.

5.7 Revising the Controller Constraints

This extra step aligns the controller constraints with the loss scenarios identified, including any missing or new constraints and complementing constraints with missing references. In order to do that, we first map the loss scenarios to the existing controller constraints that mitigate or solve the issue. Then, we create new constraints for the scenarios without constraint or which constraint does not fully satisfy the scenario.

In total, we identified 22 new safety-related constraints, where 3 constraints were gathered from other documents (explained before), and 19 new constraints were generated from loss scenarios. The new constraints are detailed in Table 5.1 and listed in Appendix D, extracted from FASTEN.

Constraint	**Rationale**
CC-101: Preview document using the approved document viewer before submission.	Ensures that the document is correct before submission.
CC-102: Document content is included in documents being created.	Ensures the document is complete before submission.
CC-103: When linking a document in a clinical workflow, a brief summary of the linked document is displayed.	Ensures the correct document (order) is linked to a report.
CC-104: Multiple submission of the same document is prevented.	Prevents multiple submissions of the same document.
CC-105: After starting the composition of the document, confirm to abort the document submission.	Prevents accidental abort of a document when creating/updating/cancelling.
CC-106: Errors in the submission of the document are indicated.	Ensures provider is aware of any issue during document submission.

CC-107: Users are notified if submitted documents are not timely received or any error happened during the delivery.	Enforces the sender to keep track of document delivery status, avoiding delay on patients' treatment.
CC-108: When updating or cancelling a document, the previous content is rendered.	Mitigates the update/cancel of wrong document.
CC-109: Only the latest version of a document can be updated.	Avoids missing content, updating older version of a document.
CC-110: Record target (patient) of a sent document cannot be updated.	Avoids possible errors on the receiving side if a document is matched to the wrong patient.
CC-111: When updating a document, recipients can only be added, not removed.	Ensures all original receives the updates of a document.
CC-112: Listing of new documents is performed automatically at reasonable time intervals.	Ensures new documents are timely listed.
CC-113: Get documents is performed automatically at reasonable time intervals.	Ensures new documents are timely retrieved.
CC-114: EMR system keeps the information and identification of the registered providers of the clinic up to date.	Keeps system's providers registry update.
CC-115: Receiving system is responsible for keeping track of problems in getting documents.	Facilitates provides to identify documents not correctly downloaded or not valid.
CC-116: Receiving system stores a received document only once; even the document is retrieved multiple times.	Avoids duplication if a document is downloaded multiple times by mistake.
CC-117: Receiving system allows a document to be manually matched to another patient in case it is associated to the wrong patient.	Lets providers fix documents matched to the wrong target.
CC-118: Users are notified when new versions of documents are received.	Ensures recipients (providers) are aware of document updates.

CC-119: Receiving system indicates when documents have relationships (events, orders, related docs, doc version) not fulfilled.	If an answer is orphaned, it was sent by mistake, or the original document was deleted/lost.

Table 5.1: New Controller Constraints

In addition to the two updated constraints mentioned before, we updated another four controller constraints from the first list of constraints to solve safety-concerns. Figure 5.15 show these constraints. The constraint *"CC-014"* addresses three criteria related to document version status. However, the CDA XML comprises a field to store the version of the document as an integer. Therefore use this field to ensure the ordering of the versions of the documents is safer and more straightforward than using the version status. The constraints *"CC-028"* and *"CC-030"* was updated to address the update and cancel actions. *"CC-041"* was updated to enforce the cancelling document is sent to all recipients. Finally, *"CC-045"* was updated to ensure that the get_document_status continues querying until the delivery status of the document reaches the statuses of "delivered" or "error."

With that, we conclude the hazard analysis of the eReferral workflow using the STPA adaptation. The next step would be to rephrase the new constraints as valid system requirements.

```
Req CC-014 : An existing document is replaced with the most recent version of that document
kind: controller constraint - associated control actions:
controller: EMR Document Processing - action/feedback: replace_document

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 21, 22, 24
* Updated constraint to replace documents by version, and not status.
¶
↵

Req CC-028 : Patient information is rendered when a document is being created, updated or canceled.
kind: controller constraint - associated control actions:
controller: EMR Document Entry - action/feedback: patient_included

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 41
* Updated to address update and cancel
¶
↵

Req CC-030 :
    Clinics and providers registered within CDX are rendered when a document is being created, updated or canceled.
kind: controller constraint - associated control actions:
controller: EMR Document Entry - action/feedback: list_clinics, list_providers, recipients_included
controller: CDX Registry - action/feedback: clinics, providers

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 43
* Updated to address update and cancel
¶
↵

Req CC-041 : Canceled documents are sent to all recipients of the original document and its updates.
kind: controller constraint - associated control actions:
controller: Sender Care Giver - action/feedback: cancel_document

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 57
* Updated to include all recipients of the original document and updates.
¶
↵

  Req CC-045 :
      Delivery status of sent documents are retrievable (SHOULD) and are performed until the document status is
      delivered or errored.
  kind: controller constraint - associated control actions:
  controller: EMR Document Entry - action/feedback: get_document_status, document_delivery_status
  controller: CDX Document Distribution - action/feedback: document_status

Reference: CDX Conformance Profile - CDA Level 1,
Conformance Sessions IDs 62
* Updated to ensure document delivery
¶
↵
```

Figure 5.15: Updated Controller Constraints (excerpt)

5.8 Evaluation

In order to evaluate this use case, we compared this analysis process with the previous analysis, which was performed using a manual approach with word processing and diagramming software, following these objectives:

1. The tooling should be easy to learn and use.

2. The tooling should reduce mistakes in the analysis by, for example, implementing features like verification, validation, and traceability.
3. The tooling should not create any impediment during the analysis, and if so, it should allow customization or the creation of features to remove the block.
4. The tooling should provide benefits for the analysis.

Moreover, these objectives are analyzed in relation to the quality attributes *usability*, *extensibility*, *maintainability* and *reusability*.

5.8.1 Usability

Usability is defined by the IEEE [25] as *"the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component."* However, this definition only contemplates the first evaluated objective, yet other authors and standards include attributes like efficiency, satisfaction, learnability and effectiveness in the definition of usability [68]. Thus, all four objectives can be associated with this quality attribute.

Regarding the first objective, for an analyst using FASTEN to perform the hazard analysis, the first concern is learning how to navigate within the tooling. Overall, the components are practical and straightforward, as each one addresses a separate concern. However, the user interface could be tricky to get used to because of some peculiarities of the MPS projectional editor, especially when using text templates. These concerns can happen with any software when the user has no previous experience or is not yet accustomed to the tool, including word processing.

FASTEN is built on top of Jetbrains MPS, a powerful IDE capable of managing large projects. Therefore, it is easy to organize various files in a project and browse the project files from inside the IDE. In addition, FASTEN DSLs implement some verification and complete traceability among the STPA artifacts. These features address the second objective entirely. On the other hand, regarding the manual approach, that analysis generated a large document, which can be problematic to navigate and maintain. Moreover, the traceability needs to be done manually, and there is no automatic verification of the analysis artifacts.

The third objective would be more towards missing features needed by the analysis; however, it can also be associated with usability issues that slow the work down. As mentioned before, some characteristics inherited from JetBrains MPS editor make FASTEN tricky to use, and it needs time for the user to get used to the tooling.

5.8.2 Extensibility

Extensibility is defined as *"the ease with which a system or component can be modified to increase its storage or functional capacity"* [25]. This quality attribute is essential for our study because we adapt the STPA method for our use case; thus, adjusting the tooling in the same fashion makes sense.

Recalling Section 4.2, FASTEN implements some features to support STPA. However, we found a few missing features in the tooling that we implemented as new DSLs, and we also updated a few DSLs to improve other features. Thus, fulfilling the third objective for the cases where the missing characteristics were preventing the continuity of the analysis.

The previous analysis performed in word processing does not require extensibility, as the analyst is free to use any feature available, like tables, listings, text references, images, and others. However, the lack of structured mechanisms for verification, validating, and tracing the artifacts of the analysis can lead to human mistakes and faults in the analysis.

5.8.3 Maintainability

Maintainability is about *"the ease with which a system or software component can be modified, improved, or adapted"* [25]. Those are the same aspects that refactoring techniques try to enhance.

FASTEN facilitates simple refactoring such as moving and renaming components. Regardless of those being elementary techniques even for word processing software, the linking capabilities of FASTEN allow replicating the changes across all the components using the changed element. Moreover, it is possible to implement other refactoring features if necessary, as demonstrated in Section 4.2.

5.8.4 Reusability

Reusability is defined by the IEEE [25] as *"the degree to which a software module or other work product can be used in more than one program or software system."* This definition addresses software or products, yet it can be narrowed and reused to software features or pieces of code. In this way, this attribute can help evaluate the fourth objective of the analysis.

FASTEN facilitates the reuse of components of the hazard analysis through the

linking capabilities among components within the analysis and reuse of parts of components in other parts of the analysis, as mentioned in the implementation of the new Loss Scenarios in Section 4.2. In addition, the base components of the STPA, such as losses and hazards tables, and control loop diagrams, can be easily reutilized to analyze other scenarios within the same FASTEN project or even be copied to other projects.

Chapter 6

Conclusion

This thesis examines the application of an adaptation of the STPA method on interoperability conformance profiles for the CDX system. This hazard analysis aims to assess and improve the safety of the CDX system interoperability by ensuring that the CDX Conformance Profiles cover all potential safety concerns found by the analysis. Moreover, the analysis is performed using FASTEN, a specialized software for system engineering of critical systems.

The interoperability provided from CDX to the EMR systems supports clinical workflows, which assists clinical decisions. Therefore, any safety-related issue with this socio-technical system, formed by clinical providers, EMR systems and CDX, can put patients' health at risk, ergo the importance of performing a hazard analysis on this system.

We opted for the STPA method for performing the hazard analysis because STPA is grounded on systems theory and, therefore, more suitable for a socio-technical system than traditional hazard analysis methods. In addition, we adapted the STPA method to accommodate better specification documents — CDX Conformance Profile and CDX Technical Overview — as input to create lists of controller constraints for the system infrastructure and conformance target. Another modification we made was to refine the controller constraints after having the lost scenarios identified, albeit the original STPA advises to perform this step before the loss scenarios identification.

This same STPA adaptation was applied before for the eReferral workflow using the CDX system interoperability [66, 67], but only evaluating the one part (ordering) of the workflow. This thesis complements the previous study by evaluating the whole eReferral workflow, i.e., ordering and reporting. In addition, instead of using word processing software, we utilized FASTEN to support the analysis in this work. We

extended some of the FASTEN DSLs in order to facilitate and cover the different situations we encountered during the analysis. For instance, we implemented a new language structure to handle the loss scenarios according to the STPA guidelines; and we extended the DSL for requirements to handle better the controller constraints.

6.1 Results

The hazard analysis of the eReferral workflow resulted in finding 22 new or missing safety constraints and updating 7 existing constraints. This result is similar to the previous study, which was performed only with the ordering part of the eReferral workflow. However, including the reporting part of the eReferral workflow in the analysis gives us more context of the whole scenario and, therefore, a better perception of what could go wrong when creating and receiving reports.

We identified three constraints related to the association of documents in a clinical workflow, where *"CC-043: Documents in a clinical workflow are linked together."* was extracted from the CDX Conformance Profile, and *"CC-103: When linking a document in a clinical workflow, a brief summary of the linked document is displayed."* and *"CC-109: Receiving system indicates when documents have relationships (events, orders, related docs, doc version) not fulfilled."* were created to address UCAs.

FASTEN has proven to be a valuable tool for conducting STPA hazard analysis, even not being a production-ready tool. FASTEN's collection of DSLs helps create and keep track of the different STPA artifacts. In addition, the relationships among languages concepts make it simple to navigate and find things without needing searching tools or looking up visually. Moreover, the extension capabilities of Jetbrains MPS facilitate the customization of the implemented DSLs and the addition of new languages or features if needed. Jetbrains MPS is a powerful workbench for DSL design, and it has comprehensive documentation. Furthermore, The Jetbrains MPS's projectional editor helps the DSLs crafting because it edits the AST directly.

6.2 Limitations

Although the hazard analysis can ensure that an electronic clinical workflow is safe, some critical scenarios still can occur because events outside the system's boundaries cause them. For example, in the following scenario, the patient's health is in peril

because the primary care (human part of the system) relies on the system to perform any further action.

Scenario: The primary care provider sends an order for a second care provider (specialist) to a specific clinic. However, the specialist does not work at that clinic anymore, and the registry in CDX is not updated. The clinic receives the order, and the document is deleted because the provider does not work there and is no more active in the EMR system. After a while, the primary care provider sees in the system that the order was delivered, but in fact, it was not. Then, believing the order has been delivered, the primary care provider postpones treating the patient because they need a response from the specialist.

This hazard analysis was performed by only one analyst and backed up by previous work performed by a different analyst. This is a limitation because it is recommended that a multidisciplinary perform the hazard analysis to have different points of view of the problem. Moreover, the comparison between manual and software-supported approaches was backed up by only one analyst performing the two modes of analysis. This is also a limitation because it can be biased by personal preferences and experiences.

The tooling, FASTEN, also presents some caveats. For instance, the lack of documentation or embedded help and other aid features, typical for commercial tools, makes the learning process challenging. Moreover, learning the Jetbrains MPS tooling is not easy, especially for someone unfamiliar with language design. Another limitation is related to the projectional editor. It takes a bit to get used to it because this editor does not have the same fluidity as a regular text editor.

6.3 Future Work

In this study, we performed the STPA hazard analysis in a real conformance profile to verify that this is complete in a safe sense. As observed, the analysis produced some valid recommendations for improving the safety aspect of the conformance profile. It may be worth investigating the implications of creating safety-concerned conformance profiles straight from the system specification using STPA.

This work presents a limited extension on evaluating the benefits of using a specialized tool for the STPA hazards analysis since the same analyst performed both approaches. A further study using separate teams performing each method would possibly provide a better comparison.

FASTEN is a tool under constant development, so it has many parts that can be improved. We list a few suggestions based on our experience with this tool in the following paragraphs.

FASTEN implements a few verifications for the consistency/completeness of the analysis. For example, the UCAs table shows a warning when control actions of a controller are not included in the table. Other similar verifications can be implemented, for example, checking if all UCAs are addressed by loss scenarios.

The links for other elements (e.g. the hazards addressed by a UCA) could show more information about the linked element in a popup or tooltip, so the user does not need to navigate to other files to recall the content of that element.

The UCAs can be textual or follow a pre-formatted pattern composed of context and consequence. This pattern could not fit for every hazard analysis, so a customized pattern for UCAs and other elements, such as safety constraints, could be helpful cases where the analyst uses the same wording for multiple UCAs.

In the new loss scenarios document, we implemented a feature to copy the content of the UCA linked to the loss scenario to facilitate writing down the loss scenario content. The same approach can be used to implement similar features for the safety constraints or fill the UCA templates suggested above.

Bibliography

[1] Asim Abdulkhaleq and Stefan Wagner. XSTAMPP : An eXtensible STAMP Platform As Tool Support for Safety Engineering. *STAMP Conference*, pages 2–5, 2015. URL: http://elib.uni-stuttgart.de/handle/11682/3550.

[2] Asim Abdulkhaleq and Stefan Wagner. XSTAMPP 2.0: new improvements to XSTAMPP Including CAST accident analysis and an extended approach to STPA. In *STAMP Workshop (5th, 2016, Cambridge)*, 2016. URL: https://elib.uni-stuttgart.de/handle/11682/8766.

[3] Carlos Agostinho, Yves Ducq, Gregory Zacharewicz, João Sarraipa, Fenareti Lampathaki, Raul Poler, and Ricardo Jardim-Goncalves. Towards a sustainable interoperability in networked enterprise information systems: Trends of knowledge and model-driven technology. *Computers in Industry*, 79:64–76, jun 2016. doi:10.1016/j.compind.2015.07.001.

[4] Juncal Alonso, Iker Martínez de Soria, Leire Orue-Echevarria, and Mikel Vergara. Enterprise Collaboration Maturity Model (ECMM): Preliminary Definition and Future Challenges. In *Enterprise Interoperability IV*, pages 429–438. Springer London, 2010. doi:10.1007/978-1-84996-257-5_40.

[5] Esra Bas. STPA methodology in a socio-technical system of monitoring and tracking diabetes mellitus. *Applied Ergonomics*, 89:103190, nov 2020. doi:10.1016/j.apergo.2020.103190.

[6] Tim Benson and Grahame Grieve. Why Interoperability Is Hard. In *Principles of Health Interoperability*, Health Information Technology Standards, pages 19–35. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-30370-3_2.

[7] C4ISR AWG. Levels of information systems interoperability (LISI). Technical Report 30 March 1998, DoD, Washington, DC, mar 1998.

[8] CAIRIS. CAIRIS. Accessed: 2021-02-13. URL: https://cairis.org/.

[9] CAIRIS. CAIRIS Documentation. Accessed: 2021-02-13. URL: https://cairis.readthedocs.io/en/latest/index.html.

[10] CAIRIS. CAIRIS GitHub repository. Accessed: 2022-01-10. URL: https://github.com/cairis-platform/cairis.

[11] J Chapman, B Lott, and A Bruce. Clinical Document Exchange-Technical Overview, 2016.

[12] David Chen, Guy Doumeingts, and François Vernadat. Architectures for enterprise integration and interoperability: Past, present and future. *Computers in Industry*, 59(7):647–659, sep 2008. doi:10.1016/j.compind.2007.12.016.

[13] Robert H. Dolin, Liora Alschuler, Sandy Boyer, Calvin Beebe, Fred M. Behlen, Paul V. Biron, and A. Shabo (Shvo). HL7 Clinical Document Architecture, Release 2. *Journal of the American Medical Informatics Association*, 13(1):30–39, jan 2006. doi:10.1197/jamia.M1888.

[14] Clifton A Ericson. *Hazard analysis techniques for system safety.* John Wiley \& Sons, 2005.

[15] European Commission. *New European Interoperability Framework: Promoting seamless services and data flows for European public administrations. European Union.* Publications Office, nov 2017. doi:10.2799/78681.

[16] Thomas C Ford, John M Colombi, Scott R Graham, and David R Jacques. A Survey on Interoperability Measuremen. In *Twelfth International Command and Control Research and Technology Symposium (12th ICCRTS)*, Newport, RI, jun 2007. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH. URL: https://apps.dtic.mil/docs/citations/ADA481314.

[17] Martin Fowler. ProjectionalEditing, 2008. Accessed: 2021-02-11. URL: https://martinfowler.com/bliki/ProjectionalEditing.html.

[18] Wided Guédria, Yannick Naudet, and David Chen. Maturity model for enterprise interoperability. *Enterprise Information Systems*, 9(1):1–28, jan 2015. doi:10.1080/17517575.2013.805246.

[19] Didem Gürdür and Fredrik Asplund. A systematic review to merge discourses: Interoperability, integration and cyber-physical systems. *Journal of Industrial Information Integration*, 9:14–23, mar 2018. doi:10.1016/j.jii.2017.12.001.

[20] HIMSS. Interoperability in Healthcare - HIMSS, feb 2013. Accessed: 2021-05-09. URL: https://www.himss.org/resources/interoperability-healthcare.

[21] Celso Massaki Hirata, Fellipe Guilherme Rey de Souza, Rodrigo Martins Pagliares, Juliana de Melo Bezerra, Filipe Parisoto Ribeiro, and João Hugo Marinho Maimone. WebSTAMP. Accessed: 2021-02-12. URL: http://webstamp.herokuapp.com/.

[22] HL7. About Health Level Seven International, 2013. Accessed: 2021-01-20. URL: http://www.hl7.org/about/index.cfm?ref=nav.

[23] Erik Hollnagel. The changing nature of risk. *Ergonomics Australia Journal*, 22(1-2):33–46, 2008.

[24] Erik Hollnagel. *FRAM, the functional resonance analysis method: modelling complex socio-technical systems*. Ashgate Publishing, Ltd., 2012.

[25] IEEE. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. *IEEE Std 610*, pages 1–217, 1991. URL: https://ieeexplore.ieee.org/document/182763, doi:10.1109/IEEESTD.1991.106963.

[26] Infomation-technology Promotion Agency. STAMP Workbench. Accessed: 2021-02-12. URL: https://www.ipa.go.jp/english/sec/reports/20180330.html.

[27] Interior Health Authority. CDX - Clinical Document eXchange. Accessed: 2021-02-13. URL: https://bccdx.ca/Pages/default.aspx.

[28] Joseph Kaberuka and Christopher Johnson. Adapting STPA-sec for Socio-technical Cyber Security Challenges in Emerging Nations: A Case Study in Risk Management for Rwandan Health Care. In *2020 International Conference*

on Cyber Security and Protection of Digital Services (Cyber Security), pages 1–9. IEEE, jun 2020. doi:10.1109/CyberSecurity49315.2020.9138863.

[29] Herbert Kubicek, Ralf Cimander, and Hans Jochen Scholl. Layers of Interoperability. In *Organizational Interoperability in E-Government*, pages 85–96. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-22502-4_7.

[30] Wen-Shing Lee, Doris L Grosh, Frank A Tillman, and Chang H Lie. Fault tree analysis, methods, and applications — a review. *IEEE transactions on reliability*, 34(3):194–203, 1985.

[31] Nancy Leveson. A new accident model for engineering safer systems. *Safety Science*, 42(4):237–270, apr 2004. doi:10.1016/S0925-7535(03)00047-X.

[32] Nancy G. Leveson. Safety as a system property. *Communications of the ACM*, 38(11):146, nov 1995. doi:10.1145/219717.219816.

[33] Nancy G. Leveson. *Engineering a Safer World*. The MIT Press, 2012. doi:10.7551/mitpress/8179.001.0001.

[34] Nancy G. Leveson and John P. Thomas. *STPA Handbook*. 2018. URL: http://psas.scripts.mit.edu/home/get{_}file.php?name=STPA{_}handbook.pdf.

[35] Nancy G. Leveson and Kathryn Anne Weiss. Software System Safety. In *Safety Design for Space Systems*, pages 475–505. Elsevier, 2009. doi:10.1016/B978-0-7506-8580-1.00015-4.

[36] Alan Bruce Lorraine Constable, Cynthia Robertson, Jeremy Chapman. British Columbia CDA Implementation Guide - Version 4.0, 2016. URL: https://www2.gov.bc.ca/assets/gov/health/practitioner-pro/bc-ehr-cda-implementation-guide.pdf.

[37] Rausand Marvin Rausand, Barros Anne Barros, and Hoyland Arnljot Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications*. Wiley-Blackwell, 2020.

[38] Fieran Mason-Blakley. *Information System Hazard Analysis*. PhD thesis, University of Victoria, 2017.

[39] Mbeddr. FASTEN GitHub repository. Accessed: 2022-01-10. URL: https://github.com/mbeddr/mbeddr.formal.

[40] Kai Mindermann, Frederik Riedel, Asim Abdulkhaleq, Christoph Stach, and Stefan Wagner. Exploratory Study of the Privacy Extension for System Theoretic Process Analysis (STPA-Priv) to Elicit Privacy Risks in eHealth. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 90–96. IEEE, sep 2017. arXiv:1710.11571, doi:10.1109/REW.2017.30.

[41] Jillian Oderkirk. Readiness of electronic health record systems to contribute to national health information and research. *OECD Health Working Papers No. 99*, (99), 2017. URL: https://www.oecd-ilibrary.org/content/paper/9e296bf3-en.

[42] Frank Oemig and Robert Snelick. *Principles of Conformance Testing*, pages 383–444. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-44839-8_11.

[43] Jinsoo Park and Sudha Ram. Information systems interoperability: What Lies Beneath? *ACM Transactions on Information Systems*, 22(4):595–632, oct 2004. doi:10.1145/1028099.1028103.

[44] Todd Pawlicki, Aubrey Samost, Derek W. Brown, Ryan P. Manger, Gwe-Ya Kim, and Nancy G. Leveson. Application of systems and control theory-based hazard analysis to radiation oncology. *Medical Physics*, 43(3):1514–1530, mar 2016. doi:10.1118/1.4942384.

[45] Jens Rasmussen. Risk management in a dynamic society: a modelling problem. *Safety science*, 27(2-3):183–213, 1997.

[46] Daniel Ratiu. FASTEN - Formal Specification Environment. Accessed: 2021-02-11. URL: https://sites.google.com/site/fastenroot/home.

[47] Daniel Ratiu, Marco Gario, and Hannes Schoenhaar. FASTEN: An Open Extensible Framework to Experiment with Formal Specification Approaches. In *2019 IEEE/ACM 7th International Conference on Formal Methods in Software Engineering (FormaliSE)*, pages 41–50. IEEE, IEEE, may 2019. doi:10.1109/FormaliSE.2019.00013.

[48] Daniel Ratiu, Arne Nordmann, Peter Munk, Carmen Carlan, and Markus Völter. FASTEN: An Extensible Platform to Experiment with Rigorous Modeling of Safety-Critical Systems. In *Domain-Specific Languages in Practice*, pages 131–164. Springer, 2021. URL: https://link.springer.com/chapter/10.1007/978-3-030-73758-0{_}5, doi:10.1007/978-3-030-73758-0_5.

[49] Reza Rezaei, Thiam Kian Chiew, and Sai Peck Lee. An interoperability model for ultra large scale systems. *Advances in Engineering Software*, 67:22–46, jan 2014. doi:10.1016/j.advengsoft.2013.07.003.

[50] C Robertson, J Martens, J DeLeenheer, and I Collet. CDX Conformance Profile 001 EMR System Conformance CDA Level 1, 2017.

[51] Aubrey Samost. *A systems approach to patient safety: preventing and predicting medical accidents using systems theory*. PhD thesis, Massachusetts Institute of Technology, 2015.

[52] Kamran Sartipi and Azin Dehmoobad. Cross-domain information and service interoperability. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services - iiWAS '08*, iiWAS '08, page 25, New York, New York, USA, 2008. ACM Press. doi:10.1145/1497308.1497318.

[53] Sasha Bojicic. Pan-Canadian Transport Level Interoperability Specification, Version 3.1. Technical report, Canada Health Infoway, Toronto, 2011. URL: https://infocentral.infoway-inforoute.ca/en/resources/docs/hl7/implementation-hl7/implementation-exchange/1733-transport-level-interoperability.

[54] Natalia Silvis-Cividjian, Wilko Verbakel, and Marjan Admiraal. Using a systems-theoretic approach to analyze safety in radiation therapy-first steps and lessons learned. *Safety Science*, 122:104519, feb 2020. doi:10.1016/j.ssci.2019.104519.

[55] Empirical software engineering research group. SE-Stuttgart/XSTAMPP GitHub repository. Accessed: 2022-01-10. URL: https://github.com/SE-Stuttgart/XSTAMPP.

[56] Fellipe G.R. Souza, Daniel P. Pereira, Rodrigo M. Pagliares, Simin Nadjm-Tehrani, and Celso M. Hirata. WebSTAMP: a Web Application for STPA & STPA-Sec. *MATEC Web of Conferences*, 273:02010, feb 2019. doi:10.1051/matecconf/201927302010.

[57] Sardar Muhammad Sulaman, Armin Beer, Michael Felderer, and Martin Höst. Comparison of the FMEA and STPA safety analysis methods–a case study. *Software Quality Journal*, 27(1):349–387, mar 2019. doi:10.1007/s11219-017-9396-0.

[58] John Thomas. *Extending and Automating STPA for Requirements Generation and Analysis*. PhD thesis, Massachusetts Institute of Technology, 2013.

[59] Andreas Tolk and James A Muguira. The Levels of Conceptual Interoperability Model. *Fall Simulation Interoperability Workshop*, 7(September):1–9, aug 2003.

[60] Andreas Tolk, Charles D. Turnitsa, Saikou Y. Diallo, and Leslie S. Winters. Composable M&S Web Services for Net-Centric Applications. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 3(1):27–44, jan 2006. doi:10.1177/875647930600300104.

[61] Peter Underwood and Patrick Waterson. A critical review of the STAMP, FRAM and Accimap systemic accident analysis models. *Advances in human aspects of road and rail transportation*, pages 385–394, 2012.

[62] Hans Van Der Veer and Anthony Wiles. Achieving Technical Interoperability: the ETSI Approach. *European Telecommunications Standards Institute*, (3):29, 2008. URL: https://portal.etsi.org/CTI/Downloads/ETSIApproach/IOPwhitepaperEdition3final.pdf.

[63] François B. Vernadat. Technical, semantic and organizational issues of enterprise interoperability and networking. *Annual Reviews in Control*, 34(1):139–144, apr 2010. doi:10.1016/j.arcontrol.2010.02.009.

[64] Jéssyka Vilela, Jaelson Castro, Luiz Eduardo G Martins, and Tony Gorschek. Integration between requirements engineering and safety analysis: A systematic literature review. *Journal of Systems and Software*, 125:68–92, mar 2017. doi:10.1016/j.jss.2016.11.031.

[65] Alan Wassyng, Tom Maibaum, and Mark Lawford. On Software Certification: We Need Product-Focused Approaches. In Christine Choppy and Oleg Sokolsky, editors, *Foundations of Computer Software. Future Trends and Techniques for Development*, pages 250–274, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. doi:10.1007/978-3-642-12566-9_13.

[66] Jens H Weber and Oscar Costa. Hazard Analysis of Interoperability Conformance Profiles. In *CASCON '19: Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pages 156–165. IBM Corp., 2019. URL: https://dl.acm.org/doi/abs/10.5555/3370272.3370289.

[67] Jens H Weber and Oscar Costa. Adapting a System-Theoretic Hazard Analysis Method for the Analysis of an eHealth Interoperability Conformance Profile. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2020:693–702, 2020. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC7233088.

[68] Paweł Weichbroth. Usability attributes revisited: a time-framed knowledge map. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 1005–1008. IEEE, 2018.

[69] Abouzar Yousefi, Manuel Rodriguez Hernandez, and Valentin Lopez Peña. Systemic accident analysis models: A comparison study between AcciMap, FRAM, and STAMP. *Process Safety Progress*, 38(2):e12002, 2019.

[70] Nanda Anugrah Zikrullah, Hyungju Kim, Meine JP van der Meulen, Gunleiv Skofteland, and Mary Ann Lundteigen. A comparison of hazard analysis methods capability for safety requirements generation. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and ReliabilityJournal of Risk and Reliability*, 2021.

Appendix A

Controller Constraints

3_Controller Constraints

model cdx.interoperability

Req CC-001 : Standardized documents are received.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** get_document
controller: CDX Document Retrieval **- action/feedback:** document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Session IDs 1, 2, 3, 4, 38

Req CC-002 : Standardized documents are rendered.

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** open_document
controller: EMR Document Processing **- action/feedback:** document_rendered
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 1, 2, 3, 4, 28, 38

Req CC-003 : Received documents are automatically assigned to at least one provider.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** auto_assign_provider
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 12, 13

Req CC-004 : Received documents can be manually assigned to a provider (SHOULD).

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** assign_provider
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 12, 13

Req CC-005 : Received documents are not automatically deleted when no assigned to a provider.

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** assign_provider
controller: EMR Document Processing **- action/feedback:** auto_assign_provider
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 12, 13

Req CC-006 : Received documents are automatically matched to an existing patient using the 4 point matching criteria.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** get_patient patient_matched
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 14

Req CC-007 : Users are notified when a received document can not be matched to a patient using the 4 point matching criteria.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** patient_matched
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 15

Req CC-008 : Patient demographic data can be manually updated to match a received document (SHOULD).

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** update_patient
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 15

Req CC-009 : Patient can be manually created to match a received document.

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** create_patient
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 16

Req CC-010 : Patient can be manually matched to a received document.

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** match_patient
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 16

Req CC-011 : Received documents can be manually deleted when delivered to clinic by mistake.

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** delete_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 17

Req CC-012 : Deleted documents can be manually restored after deletion (SHOULD).

kind: controller constraint - associated control actions:
controller: Receiver Care Giver **- action/feedback:** restore_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 17

Req CC-013 : Status of received documents are indicated in the user interface.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_status
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 18, 19, 20

Req CC-014 : An existing document is replaced with the most recent version of that document.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** replace_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 21, 22, 24

- Updated constraint to replace documents by version, and not status.

Req CC-015 : History of replaced documents are maintained.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** retain_replaced_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 22, 24

Req CC-016 : Users are notified when cancelled documents are received (MAY).

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_canceled
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 23

- Updated to SHOULD constraint

Req CC-017 : Documents are rendered in an approved document viewer.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_rendered
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 25

Req CC-018 : Received documents that are new (not reviewed) are indicated in the user interface.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_visualized
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 26

Req CC-019 : Previous versions of a received document are retained for visualization.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** retain_replaced_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 27

Req CC-020 : Presence and number of attachments in a received documents are indicated in the user interface.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_attachments
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 30

Req CC-021 : Attachments of specified formats are rendered.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_attachments
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 31

Req CC-022 : Attachments are stored in patient chart.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** get_document store_attachment
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 32

Req CC-023 : Sent documents are limited in size.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 34

Req CC-024 : Attachments being send are restrict to specified formats.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 33, 35

Req CC-025 : Attachments being send are secured by cryptographic hash function.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document

- Reference: CDA XML Schema.
- Missing constraint in conformance profile!

Req CC-026 : Standardized documents are sent.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 39

Req CC-027 : Sending documents have approved template IDs and LOINC codes.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 40, 56

Req CC-028 : Patient information is rendered when a document is being created, updated or canceled.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** patient_included
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 41

- Updated to address update and cancel

Req CC-029 : Patient information (at least 4-matching-point) is included in documents being created.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 42

Req CC-030 : Clinics and providers registered within CDX are rendered when a document is being created, updated or canceled.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** list_clinics list_providers

recipients_included
controller: CDX Registry **- action/feedback:** clinics providers
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 43

- Updated to address update and cancel

Req CC-031 : Clinics and providers registered within CDX are included in documents being created

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 44, 46, 47

Req CC-032 : Documents successfully delivered to the CDX system are indicated.

kind: controller constraint - associated control actions:
controller: CDX Document Distribution **- action/feedback:** submission_ack
controller: EMR Document Entry **- action/feedback:** document_sent
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 44

Req CC-033 : Creation date is included in sent documents.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 48

Req CC-034 : Received data is included in received documents.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** get_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 50

Req CC-035 : Sent date is included in sent documents.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** submit_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 51

Req CC-036 : Sent documents can be updated by the sender (SHOULD).

kind: controller constraint - associated control actions:
controller: Sender Care Giver **- action/feedback:** update_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 55

Req CC-037 : Updated information is clearly identifiable in the documents (SHOULD).

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_rendered
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 55

Req CC-038 : Updated documents are linked to their parent documents.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** replace_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 55

Req CC-039 : Historical versions of documents are accessible.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_history
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 55

Req CC-040 : Sent documents can be canceled by the sender (MAY).

kind: controller constraint - associated control actions:
controller: Sender Care Giver **- action/feedback:** cancel_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 57

- Updated to SHOULD constraint

Req CC-041 : Canceled documents are sent to all recipients of the original document and its updates.

kind: controller constraint - associated control actions:
controller: Sender Care Giver **- action/feedback:** cancel_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 57

- Updated to include all recipients of the original document and updates.

Req CC-042 : Support documents are referred in CDA documents.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** documents_referred
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 58, 59

Req CC-043 : Documents in a clinical workflow are linked together.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** document_linked
controller: EMR Document Processing **- action/feedback:** link_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 60

Req CC-044 : Documents can be queried by IDs.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** search_document
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 61

Req CC-045 : Delivery status of sent documents are retrievable (SHOULD) and are performed until the document status is delivered or errored.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** get_document_status document_delivery_status
controller: CDX Document Distribution **- action/feedback:** document_status
Reference: CDX Conformance Profile - CDA Level 1, Conformance Sessions IDs 62

- Updated to ensure document delivery

Req CC-046 : CDX routes documents to all locations (clinics) associated with the intended recipients (provider)

present in the document if no clinic is specified. **kind:** controller constraint - associated control actions:
controller: CDX Document Distribution **- action/feedback:** route_document
Reference: CDX Technical Overview, Section 3.2.1

Req CC-047 : CDX routes documents to the specific locations (clinics) present in the document,

regardless of whether the providers are specified and are not associated to the specified clinics. **kind:** controller constraint - associated control actions:
controller: CDX Document Distribution **- action/feedback:** route_document
Reference: CDX Technical Overview, Section 3.2.2

Req CC-048 : CDX allows providers with multiple types of unique identifiers.

kind: controller constraint - associated control actions:
controller: CDX Registry **- action/feedback:** providers
Reference: CDX Technical Overview, Section 3.3

Req CC-049 : CDX uses only unique CDX identifiers for clinics.

kind: controller constraint - associated control actions:
controller: CDX Registry **- action/feedback:** clinics

Reference: CDX Technical Overview, Section 3.4

Req CC-050 : CDX validates the documents when they are received.

kind: controller constraint - associated control actions:
controller: CDX Document Distribution **- action/feedback:** submission_ack
Reference: CDX Technical Overview, Section 3.5.1

Req CC-051 : Acknowledgment response errors are used for debugging purposes.

kind: controller constraint - associated control actions:
controller: CDX Document Distribution **- action/feedback:** submission_ack
Reference: CDX Technical Overview, Section 3.5.1.1

Req CC-052 : CDX system only make the documents available for their recipients.

kind: controller constraint - associated control actions:
controller: CDX Document Distribution **- action/feedback:** route_document
Reference: CDX Technical Overview, Section 3.5.2

Req CC-053 : CDX system lists CDA documents as new documents only until the recipient/location requests the document.

kind: controller constraint - associated control actions:
controller: CDX Document Retrieval **- action/feedback:** new_documents
Reference: CDX Technical Overview, Section 3.5.2.1

Req CC-054 : Receiving system is responsible for keeping track of the listing of new documents.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** list_new_documents
controller: CDX Document Retrieval **- action/feedback:** new_documents
Reference: CDX Technical Overview, Section 3.5.2.1

- Missing constraint in conformance profile!

Req CC-055 : Receiving system is responsible for download and store documents from CDX.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** get_document
controller: CDX Document Retrieval **- action/feedback:** document
Reference: CDX Technical Overview, Section 3.5.2.1

- Missing constraint in conformance profile!

Req CC-056 : Submitted documents are available in the CDX system for searching for a finite time.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** search_document
controller: CDX Document Retrieval **- action/feedback:** document
Reference: CDX Technical Overview, Section 3.5.2.2

Req CC-057 : Registered clinics and providers are searchable in the CDX system.

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** list_clinics list_providers
controller: CDX Registry **- action/feedback:** clinics providers
Reference: CDX Technical Overview, Section 3.5.3

Req CC-058 : CDX provides a secure channel to transfer documents.

kind: controller constraint - associated control actions:
controller: CDX Document Distribution **- action/feedback:** route_document
Reference: CDX Technical Overview, Section 3.6.3

Appendix B

UCAs

4_UCAs

model cdx.interoperability

Unsafe Control Analysis for Control Structure: 2.2_Document Exchange Control Structure

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazad	Too Soon/Late, Out of Sequence	Stopped Too Soon, Applied Too Long
Sender Care Giver	create_document		UCA-create_document-provided_with_wrong_target @controller(Sender Care Giver) provides @action(create_document) with the wrong target information [H-01] UCA-create_document-provided_with_wrong_recipient @controller(Sender Care Giver) provides @action(create_document) with the wrong recipient (provider or clinic) [H-02 H-04] UCA-create_document-provided_with_wrong_content @controller(Sender Care Giver) provides @action(create_document) with the wrong content [H-03] UCA-create_document-provided_for_wrong_order @controller(Sender Care Giver) provides @action(create_document)(report) for the wrong document (order) [H-06]	N/A	UCA-create_document-provided_repeatedly @controller(Sender Care Giver) provides repeatedly @action(create_document) for the same document [H-05] UCA-create_document-stopped_too_soon @controller(Sender Care Giver) stop providing @action(create_document) before @controller(EMR Document Entry) conclude @action(submit_document) [H-01 H-02 H-03 H-04]

Table B.1 continued from previous page

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazad	Too Soon/Late, Out of Sequence	Stopped Too Soon, Applied Too Long
	update_document	N/A	UCA-update_document-provided_for_wrong_document @controller(Sender Care Giver) provides @action(update_document) for the wrong document [H-04] UCA-update_document-provided_with_wrong_target @controller(Sender Care Giver) provides @action(update_document) with the wrong target [H-01] UCA-update_document-provided_with_wrong_recipient @controller(Sender Care Giver) provides @action(update_document) with the wrong recipient (provider or clinic) [H-02 H-04] UCA-update_document-provided_with_wrong_content @controller(Sender Care Giver) provides @action(update_document) with the wrong content [H-03]	N/A	UCA-update_document-provided_repeatedly @controller(Sender Care Giver) provides repeatedly @action(update_document) for the same document [H-05] UCA-update_document-stopped_too_soon @controller(Sender Care Giver) stop providing @action(update_document) before @controller(EMR Document Entry) conclude @action(submit_document) [H-03 H-04]
	cancel_document	N/A	UCA-cancel_document-provided_for_wrong_document @controller(Sender Care Giver) provides @action(cancel_document) for the wrong document [H-04]	N/A	UCA-cancel_document_stopped_too_soon @controller(Sender Care Giver) stop providing @action(cancel_document) before @controller(EMR Document Entry) conclude @action(submit_document) [H-04]
	intervention	N/A	N/A	N/A	N/A

Table B.1 continued from previous page

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazad	Too Soon/Late, Out of Sequence	Stopped Too Soon, Applied Too Long
EMR Document Entry	submit_document	UCA-submit_document-not_provided @controller(EMR Document Entry) does not provide @action(submit_document) for a newly created, updated or cancelled document [H-04]	N/A	UCA-submit_document-provided_too_soon @controller(EMR Document Entry) provides too soon @action(submit_document) when @controller(Sender Care Giver) is still providing @action(create_document), @action(update_document) or @action(cancel_document) for a document [H-01 H-02 H-03 H-04]	UCA-submit_document-stopped_too_soon @controller(EMR Document Entry) stops too soon providing @action(submit_document) before completing the submission of a document [H-04]
	get_document_status	UCA-get_document_status-not_provided @controller(EMR Document Entry) does not provide @action(get_document_status) for a submitted document [H-04]	N/A	N/A	N/A
	get_patient	UCA-get_patient-not_provided @controller(EMR Document Entry) does not provide @action(get_patient) [H-01]	UCA-get_patient-provided_for_wrong_patient @controller(EMR Document Entry) provides @action(get_patient) for the wrong patient [H-01]	N/A	N/A
	save_patient	N/A	N/A	N/A	N/A
	link_document	UCA-link_document-not_provided_for_order @controller(EMR Document Entry) does not provide @action(link_document) for the original document (order) when creating a document (report) [H-06]	UCA-ling_document-provided_for_wrong_order @controller(EMR Document Entry) provides @action(link_document) for the wrong document (order) when submiting a document (report) [H-06]	N/A	N/A
	list_clinics	UCA-list_clinics-not_provided @controller(EMR Document Entry) does not provide @action(list_clinics) when creating/updating/caceling a document [H-02 H-04]	UCA-list_clinics-provided_for_wrong_clinic @controller(EMR Document Entry) provides @action(list_clinics) for a wrong clinic [H-02 H-04]	N/A	N/A

Table B.1 continued from previous page

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazad	Too Soon/Late, Out of Sequence	Stopped Too Soon, Applied Too Long
	list_providers	UCA-list_providers-not_provided @controller(EMR Document Entry) does not provide @action(list_providers) when creating/updating/caceling a document [H-02 H-04]	UCA-list_providers-provided_for_wrong_document @controller(EMR Document Entry) provides @action(list_providers) for the wrong document [H-02 H-04]	N/A	N/A
CDX Document Distribution	route_document	N/A	N/A	UCA-route_document-provided_out_of_sequence @controller(CDX Document Distribution) provides @action(route_document) out of sequence for two or more versions of a same document [H-04]	N/A
	get_recipients	N/A	N/A	N/A	N/A
EMR Document Processing	list_new_documents	UCA-list_new_documents-not_provided @controller(EMR Document Processing) does not provide @action(list_new_documents) [H-04]	N/A	UCA-list_new_documents-provided_too_late @controller(EMR Document Processing) provides too late @action(list_new_documents) [H-04]	N/A
	search_document	UCA-search_document-not_provided @controller(EMR Document Processing) does not provide @action(search_document) for a missed document (fail in @action(get_document)) [H-04]	N/A	N/A	N/A
	get_document	UCA-get_document-not_provided @controller(EMR Document Processing) does not provide @action(get_document) for a new document [H-04]	N/A	UCA-get_document-provided_too_late @controller(EMR Document Processing) provides too late @action(get_document) for a new document [H-04]	UCA-get_document-stopped_too_soon @controller(EMR Document Processing) stops too soon providing @action(get_document) for a new document [H-04] UCA-get_document-provided_repeatedly @controller(EMR Document Processing) provides @action(get_document) repeatedly [H-05]

Table B.1 continued from previous page

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazad	Too Soon/Late, Out of Sequence	Stopped Too Soon, Applied Too Long
	retain_replaced_document	UCA-retain_replaced_document-not_provided @controller(EMR Document Processing) does not provide @action(retain_replaced_document) after @action(replace_document) for an updated document [H-04]	N/A	N/A	N/A
	replace_document	UCA-replace_document-not_provided @controller(EMR Document Processing) does not provide @action(replace_document) for an updated document [H-05]	N/A	UCA-replace_document-provided_out_of_sequence @controller(EMR Document Processing) provides @action(replace_document) out of sequence for two or more versions of the same document [H-04]	N/A
	auto_assign_provider	UCA-auto_assign_provider-not_provided @controller(EMR Document Processing) does not provide @action(auto_assign_provider) for received document [H-04]	N/A	N/A	N/A
	auto_match_patient	UCA-auto_match_patient-not_provided @controller(EMR Document Processing) does not provide @action(auto_match_patient) for a received document [H-04]	UCA-auto_match_patient-provided_for_wrong_patient @controller(EMR Document Processing) provides @action(auto_match_patient) for the wrong patient [H-04]	N/A	N/A
	get_patient	N/A	N/A	N/A	N/A
	save_patient	N/A	N/A	N/A	N/A
	store_attachment	UCA-store_attachment-not_provided @controller(EMR Document Processing) does not provide @action(store_attachment) for a received document [H-03]	UCA-store_attachment-provided_for_wrong_patient @controller(EMR Document Processing) provides @action(store_attachment) for the wrong patient [H-03]	N/A	N/A

Table B.1 continued from previous page

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazad	Too Soon/Late, Out of Sequence	Stopped Too Soon, Applied Too Long
	link_document	UCA-link_document-not_provided_for_report @controller(EMR Document Processing) does not provide @action(link_document) for a received document (report) [H-06]	N/A	N/A	N/A
Receiver Care Giver	open_document	UCA-open_document-not_provided @controller(Receiver Care Giver) does not provide @action(open_document) for a received document [H-04]	UCA-open_document-provided_for_wrong_document @controller(Receiver Care Giver) provides @action(open_document) for the wrong document [H-04]	UCA-open_document-too_late @controller(Receiver Care Giver) provides too late @action(open_document) for a received document [H-04]	N/A
	assign_provider	N/A	UCA-assign_provider-provided_for_wrong_provider @controller(Receiver Care Giver) provides @action(assign_provider) for the wrong provider [H-04]	N/A	N/A
	match_patient	N/A	UCA-match_patient-provided_with_wrong_patient @controller(Receiver Care Giver) provides @action(match_patient) for a received document with the wrong patient [H-04]	N/A	N/A
	create_patient	UCA-create_patient-not_provided @controller(Receiver Care Giver) does not provide @action(create_patient) for a received document with new patient [H-04]	N/A	N/A	N/A
	delete_document	N/A	UCA-delete_document-provided_for_wrong_document @controller(Receiver Care Giver) provides @action(delete_document) for the wrong document [H-04]	N/A	N/A

Table B.1 continued from previous page

Source Controller	Action	Not Providing Causes Hazard	Providing Causes Hazad	Too Soon/Late, Out of Sequence	Stopped Too Soon, Applied Too Long
	restore_document	UCA-restore_document-not_provided @controller(Receiver Care Giver) does not provide @action(restore_document) for a deleted document [H-04]	UCA-restore_document-provided_for_wrong_document @controller(Receiver Care Giver) provides @action(restore_document) for a wrong deleted document [H-04]	N/A	N/A
	update_patient	UCA-update_patient-not_provided @controller(Receiver Care Giver) does not provide @action(update_patient) for a received document [H-01]	UCA-update_patient-provided_with_wrong_patient @controller(Receiver Care Giver) provides @action(update_patient) with the wrong patient [H-01]	N/A	N/A
	intervention	N/A	N/A	N/A	N/A

Appendix C

Loss Scenarios

5_Loss Scenarios Sending Documents

model cdx.interoperability

UCA: UCA-create_document-provided_with_wrong_target

Scenario: LS-SD-001.1
@controller(Sender Care Giver) provides @action(create_document) with the wrong target information when creating a document because other patients are registered in the EMR system with a similar name, or the patient changed their name and the EMR system is not up to date.

- Mitigated by @req(CC-028)

[H-01]
Scenario: LS-SD-001.2
@controller(Sender Care Giver) provides @action(create_document) with the wrong target information when creating a document because they insert the incorrect patient information into the EMR system.

- Mitigated by @req(CC-028)

[H-01]
Scenario: LS-SD-001.3

@controller(Sender Care Giver) provides @action(create_document) with the wrong target information when creating a document because they forget to select or insert the patient information in the document.

- Mitigated by @req(CC-028) and @req(CC-029)

[H-01]

UCA: UCA-create_document-provided_with_wrong_recipient

Scenario: LS-SD-002.1
@controller(Sender Care Giver) provides @action(create_document) with the wrong recipient (provider or clinic) when creating a document because other recipients (providers or clinics) have similar name or id.

- Mitigated by @req(CC-030)

[H-02 H-04]
Scenario: LS-SD-002.2
@controller(Sender Care Giver) provides @action(create_document) with the wrong recipient (provider or clinic) when creating a document because the provider changed their name and the CDX registry is not updated, or the provider is no more attending on the selected clinic and the CDX registry is not updated.

- Mitigated by @req(CC-030)

[H-02 H-04]
Scenario: LS-SD-002.3
@controller(Sender Care Giver) provides @action(create_document) with the wrong recipient (provider or clinic) when creating a document because they forget to select the recipients for the document.

- Mitigated by @req(CC-030) and @req(CC-031)

[H-02 H-04]

UCA: UCA-create_document-provided_with_wrong_content

Scenario: LS-SD-003.1
@controller(Sender Care Giver) provides @action(create_document) with the wrong content when creating a document because they get an outdated exam or clinical information from the EMR system.

- New @req(CC-101)

[H-03]
Scenario: LS-SD-003.2
@controller(Sender Care Giver) provides @action(create_document) with the wrong content when creating a document because they forget to select the clinical information from the EMR system or to insert the document content.

- New @req(CC-101) and @req(CC-102)

[H-03]

UCA: UCA-create_document-provided_for_wrong_order

Scenario: LS-SD-004.1
@controller(Sender Care Giver)provides @action(create_document)(report) for the wrong document (order) because they pick the wrong order from the correct patient or the wrong patient with a similar name while creating the report.

- New @req(CC-103)

[H-06]

UCA: UCA-create_document-provided_repeatedly

Scenario: LS-SD-005.1
@controller(Sender Care Giver) provides repeatedly @action(create_document) for the same document because they do not see the submission feedback or other reasons that make them believe the document was not successfully created.

- Mitigated by @req(CC-032)
- New @req(CC-104)

[H-05]
Scenario: LS-SD-005.2
@controller(Sender Care Giver) provides repeatedly @action(create_document) for the same document because they execute the create action multiple times by mistake (ex.: double click on a web-based system) or other error (ex.: locked keyboard key or mouse button).

- New @req(CC-104)

[H-05]

UCA: UCA-create_document-stopped_too_soon

Scenario: LS-SD-006.1
@controller(Sender Care Giver) stop providing @action(create_document)because they perform some action that aborts the document creation believing that they are concluding the document creation.

- Mitigated by @req(CC-032)
- New @req(CC-105)

[H-01 H-02 H-03 H-04]
Scenario: LS-SD-006.2
@controller(Sender Care Giver)stop providing @action(create_document)because they attempt to conclude creating the document, but the document is not submitted due to some connection error, and they are not informed or do not see the error message and close the form.

- Mitigated by @req(CC-032)
- New @req(CC-106)

[H-01 H-02 H-03 H-04]

UCA: UCA-update_document-provided_for_wrong_document

Scenario: LS-SD-007.1
@controller(Sender Care Giver) provides @action(update_document) for the wrong document because they pick the wrong document from the correct patient or a patient with a similar name.

- Mitigated by @req(CC-028), @req(CC-030), @req(CC-037) and @req(CC-038)
- New @req(CC-108)

[H-04]
Scenario: LS-SD-007.2
@controller(Sender Care Giver)provides @action(update_document)for an outdated version of the correct document, losing the information contained in the newest versions.

- New @req(CC-109)

[H-03]

UCA: UCA-update_document-provided_with_wrong_target

Scenario: LS-SD-008.1
@controller(Sender Care Giver)provides @action(update_document)with a different target when updating a document because they notice that the original target was incorrect. The receiver system will update the document, but the document will continue linked to the original target since it is an update of an existing document.

- New @req(CC-110)

[H-01]

UCA: UCA-update_document-provided_with_wrong_recipient

Scenario: LS-SD-009.1
@controller(Sender Care Giver)provides @action(update_document)with different recipients (provider or clinic), missing some of the original recipients.

- New @req(CC-111)

[H-02 H-04]

UCA: UCA-update_document-provided_with_wrong_content

Scenario: LS-SD-010.1
@controller(Sender Care Giver) provides @action(update_document) with wrong content when updating a document because they get and outdated exam or clinical inforrmation from the EMR system.

- Mitigated by @req(CC-037) and @req(CC-039)
- New @req(CC-101)

[H-03]
Scenario: LS-SD-010.2
@controller(Sender Care Giver) provides @action(update_document) with wrong content when updating a document because they forget to update the document content.

- Mitigated by @req(CC-037) and @req(CC-039)
- New @req(CC-101)

[H-03]

UCA: UCA-update_document-provided_repeatedly

Scenario: LS-SD-011.1
@controller(Sender Care Giver) provides repeatedly @action(update_document) for the same document because they do not see the submission feedback or other reasons that make them believe the document was not successfully updated.

- New @req(CC-104)

[H-05]
Scenario: LS-SD-011.2
@controller(Sender Care Giver) provides repeatedly @action(update_document) for the same document because they execute the update action multiple times by mistake (ex.: double click on a web-based system) or other error (ex.: locked keyboard key or mouse button).

- New @req(CC-104)

[H-05]

UCA: UCA-update_document-stopped_too_soon

Scenario: LS-SD-012.1
@controller(Sender Care Giver) stop providing @action(update_document) because they perform some action that aborts the document updating believing that they are concluding the document updating.

- Mitigated by @req(CC-032)
- New @req(CC-105)

[H-03 H-04]
Scenario: LS-SD-012.2
@controller(Sender Care Giver)stop providing @action(update_document)because they attempt to conclude the document update, but the document is not submitted due to some connection error, and they are not informed or do not see the error message and close the form.

- Mitigated by @req(CC-032)
- New @req(CC-106)

[H-03 H-04]

UCA: UCA-cancel_document-provided_for_wrong_document

Scenario: LS-SD-013.1
@controller(Sender Care Giver) provides @action(cancel_document) for the wrong document because they pick the wrong document to cancel.

- Mitigated by @req(CC-028), @req(CC-030)
- New @req(CC-108)

[H-04]
Scenario: LS-SD-013.2
@controller(Sender Care Giver)provides @action(cancel_document)for the wrong document because they execute the cancel action by mistake.

- Mitigated by @req(CC-028), @req(CC-030)
- New @req(CC-108)

[H-04]

UCA: UCA-cancel_document_stopped_too_soon

Scenario: LS-SD-014.1
@controller(Sender Care Giver) stop providing @action(cancel_document) because they perform some action that aborts the cancel action believing that they are canceling the document.

- Mitigated by @req(CC-032)

- New @req(CC-105)

[H-04]
Scenario: LS-SD-014.2
@controller(Sender Care Giver)stop providing@action(cancel_document)because they attempt to conclude cancelling the document, but the document is not submitted due to some connection error, and they are not informed or do not see the error message and close the form.

- Mitigated by @req(CC-032)
- New @req(CC-106)

[H-04]

UCA: UCA-submit_document-not_provided

Scenario: LS-SD-015.1
@controller(EMR Document Entry) does not provide @action(submit_document) for a newly created, updated or cancelled document because the EMR system does not receive a command to submit the document.

- Addressed in @lossScenario(LS-SD-006.1), @lossScenario(LS-SD-012.1) and @lossScenario(LS-SD-014.1)

[H-04]
Scenario: LS-SD-015.2
@controller(EMR Document Entry) does not provide @action(submit_document) for a newly created, updated or cancelled document because there is a communication problem with the CDX system.

- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-04]

UCA: UCA-submit_document-provided_too_soon

Scenario: LS-SD-016.1
@controller(EMR Document Entry) provides too soon @action(submit_document) when @controller(Sender Care Giver) is still providing @action(create_document),@action(update_do or @action(cancel_document) for a document because the EMR System receives a command to submit the document but does not check if the document is complete before submitting it.

- Addressed in @lossScenario(LS-SD-001.3), @lossScenario(LS-SD-002.3) and @lossScenario(LS-SD-003.2)

[H-01 H-02 H-03 H-04]

UCA: UCA-submit_document-stopped_too_soon

Scenario: LS-SD-017.1
@controller(EMR Document Entry) stops too soon providing @action(submit_document) before completing the document submission because there is a communication problem with the CDX system.

- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-04]

UCA: UCA-get_document_status-not_provided

Scenario: LS-SD-018.1
@controller(EMR Document Entry)does not provide @action(get_document_status)for a submitted document, causing the @controller(Sender Care Giver) to believe that the document was delivered when it was not.

- Mitigated by @req(CC-045)

[H-04]

Control Action: EMR Document Entry - get_document_status

Scenario: LS-SD-019.1
@controller(EMR Document Entry)provides @action(get_document_status)but does not notify the @controller(Sender Care Giver) if the submitted document was delivered (due to error) or not received by the recipient, causing the SCG to believe that the document was delivered when it was not.

- New @req(CC-107)

[H-04]
Scenario: LS-SD-019.2
@controller(EMR Document Entry)provides @action(get_document_status)and notifies the @controller(Sender Care Giver)that the document as delivered. However, the clinic recipient deleted the document by mistake or because the provider no longer works there.

- Out of the scope of the CDX system. The SCG should try to contact the recipient.

[H-04]

UCA: UCA-get_patient-not_provided

Scenario: LS-SD-020.1
@controller(EMR Document Entry) does not provide @action(get_patient) when creating a document because the EMR system does not receive the command to get the patient information.

- Addressed in @lossScenario(LS-SD-003.1)

[H-01]

UCA: UCA-get_patient-provided_for_wrong_patient

Scenario: LS-SD-021.1
@controller(EMR Document Entry) provides @action(get_patient) for a wrong patient because the registered patient information is outdated or incorrect.

- Addressed in @lossScenario(LS-SD-001.1)

[H-01]

UCA: UCA-link_document-not_provided_for_order

Scenario: LS-SD-022.1
@controller(EMR Document Entry) does not provide @action(link_document) for the original document (order) when sending a document (report) because the @controller(Sender Care Giver) creates a new document instead of a report.

- Mitigated by @req(CC-043)

[H-06]

UCA: UCA-ling_document-provided_for_wrong_order

Scenario: LS-SD-023.1
@controller(EMR Document Entry)provides @action(link_document)for the wrong document (order) when submiting a document (report) because the @controller(Sender Care Giver)creates the report for the wrong order.

- Addressed by @lossScenario(LS-SD-004.1)

[H-06]

UCA: UCA-list_clinics-not_provided

Scenario: LS-SD-024.1
@controller(EMR Document Entry) does not provide @action(list_clinics) when creating/updating/caceling a document because there is a communication problem with the CDX system.

- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-02 H-04]
Scenario: Ls-SD-024.2
@controller(EMR Document Entry) does not provide @action(list_clinics) when creating/updating/caceling a document because the EMR system does not receive the command to list the clinics.

- Addressed in @lossScenario(LS-SD-002.3)

[H-02 H-04]

UCA: UCA-list_clinics-provided_for_wrong_clinic

Scenario: LS-SD-025.1
@controller(EMR Document Entry) provides @action(list_clinics) for a wrong clinic because there are other clinics registered with similar name or id.

- Addressed in @lossScenario(LS-SD-002.1)

[H-02 H-04]

UCA: UCA-list_providers-not_provided

Scenario: LS-SD-026.1
@controller(EMR Document Entry) does not provide @action(list_providers) when creating/updating/caceling a document because there is a communication problem with the CDX system.

- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-02 H-04]
Scenario: LS-SD-026.2
@controller(EMR Document Entry) does not provide @action(list_providers) when creating/updating/caceling a document because the EMR system does not receive the command to list the providers.

- Addressed in @lossScenario(LS-SD-002.3)

[H-02 H-04]

UCA: UCA-list_providers-provided_for_wrong_document

Scenario: LS-SD-027.1
@controller(EMR Document Entry) provides @action(list_providers) for the wrong document because there are other providers registered with similar name or id.

- Addressed in @lossScenario(LS-SD-002.1)

[H-02 H-04]

UCA: UCA-route_document-provided_out_of_sequence

Scenario: LS-SD-028.1
@controller(CDX Document Distribution) provides @action(route_document) out of sequence for two or more versions of a same document when one or more updates/ cancels are submitted short time after the previous document.

- Mitigated by @req(CC-014) and @req(CC-019) on reveiver side

[H-04]

Control Action: EMR Document Entry - list_clinics

Scenario: LS-SD-029.1
@controller(EMR Document Entry) provides @action(list_clinics), but the clinics are not listed due to a connection issue or error on the CDX system.

- Mitigated by @req(CC-058)
- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-02]

Control Action: EMR Document Entry - list_providers

Scenario: LS-SD-030.1
@controller(EMR Document Entry) provides @action(list_providers), but the clinics are not listed due to a connection issue or error on the CDX system.

- Mitigated by @req(CC-058)
- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-02]

Control Action: EMR Document Entry - submit_document

Scenario: LS-SD-031.1
@controller(EMR Document Entry) provides @action(submit_document), but the submission acknowledgment is not returned due to a connection issue or error on the CDX system.

- Mitigated by @req(CC-058)
- Suggestion: Resubmit a document without being sure that the previous attempt was unsuccessful will cause the document to be duplicated at the recipient side.

[H-04]

Control Action: EMR Document Entry - get_document_status

Scenario: LS-SD-032.1
@controller(EMR Document Entry) provides @action(get_document_status), but the document status is not returned due to a connection issue or error on the CDX system.

- Mitigated by @req(CC-058)
- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-04]

5_Loss Scenarios Receiving Documents

model cdx.interoperability

UCA: UCA-open_document-not_provided

Scenario: LS-RD-001.1
@controller(Receiver Care Giver) does not provide @action(open_document) for a received document because they are not notified or see the received documents in the EMR system.

- Mitigated by @req(CC-018), @req(CC-118)

[H-04]

UCA: UCA-open_document-provided_for_wrong_document

Scenario: LS-RD-002.1
@controller(Receiver Care Giver) provides @action(open_document) for the wrong document because the EMR system automatic matched the document for other patient with same name or similar information.

- Mitigated by @req(CC-006) and @req(CC-007)

[H-04]
Scenario: LS-RD-002.2
@controller(Receiver Care Giver) provides @action(open_document) for a canceled document because they are not notified that the document is canceled.

- Mitigated by @req(CC-016)

[H-04]
Scenario: LS-RD002.3
@controller(Receiver Care Giver)provides @action(open_document)for an outdated document because they are not notified that the document is updated or the system does not replace documents with a newer version.

- Mitigated by @req(CC-014) and @req(CC-118)

[H-04]

UCA: UCA-open_document-too_late

Scenario: LS-RD-003.1
@controller(Receiver Care Giver)provides too late @action(open_document)for a received document because they are not notified that the document was received.

- Mitigated by @req(CC-016), @req(CC-018) and @req(CC-118)

[H-04]

UCA: UCA-assign_provider-provided_for_wrong_provider

Scenario: LS-RD-004.1
@controller(Receiver Care Giver) provides @action(assign_provider) for the wrong provider becuse the document was addressed for the wrong clinic and there is a provider with same name or similar information in the clinic, or the document was addressed only for the clinic by mistake.

- Mitigated by @req(CC-011) when document is received by mistake

[H-04]

UCA: UCA-match_patient-provided_with_wrong_patient

Scenario: LS-RD-005.1
@controller(Receiver Care Giver) provides @action(match_patient) for a received document with the wrong patient because there are other patients with the same name or similar information.

- Mitigated by @req(CC-007)

[H-04]

UCA: UCA-create_patient-not_provided

Scenario: LS-RD-006.1
@controller(Receiver Care Giver) does not provide @action(create_patient) for a received document with new patient because the document was already matched with a existing patient that has same name or similar information.

- Mitigated by @req(CC-007)

[H-04]

UCA: UCA-delete_document-provided_for_wrong_document

Scenario: LS-RD-007.1
@controller(Receiver Care Giver) provides @action(delete_document) for the wrong document by mistake and the EMR system does not allow restore deleted documents.

- Mitigated by @req(CC-012)

[H-04]

UCA: UCA-restore_document-not_provided

Scenario: LS-RD-008.1
@controller(Receiver Care Giver) does not provide @action(restore_document) for a deleted document because the EMR system does not provide the restore action or listing deleted documents.

- Mitigated by @req(CC-012)

[H-04]

UCA: UCA-restore_document-provided_for_wrong_document

Scenario: LS-RD-009.1
@controller(Receiver Care Giver) provides @action(restore_document) for a wrong deleted document because the listing of deleted documents does not provide enough information to distinguish similar documents.

- Mitigated by @req(CC-011) and @req(CC-012)

[H-04]

UCA: UCA-update_patient-not_provided

Scenario: Ls-RD-010.1
@controller(Receiver Care Giver) does not provide @action(update_patient) for a received document because they are not notified about changes in the patient information or is not clear what are the diferences between the patient information in the document and in the EMR chart.

- Mitigated by @req(CC-007)

[H-01]

UCA: UCA-update_patient-provided_with_wrong_patient

Scenario: Ls-RD-011.1
@controller(Receiver Care Giver) provides @action(update_patient) with the wrong patient because the document is matched to another patient with same name or similar information.

- Mitigated by @req(CC-007)

[H-01]

UCA: UCA-list_new_documents-not_provided

Scenario: LS-RD-012.1
@controller(EMR Document Processing) does not provide @action(list_new_documents) because the EMR System is not configured to list new documents automatically.

- Mitigated by @req(CC-054)
- New @req(CC-112)

[H-04]

UCA: UCA-list_new_documents-provided_too_late

Scenario: LS-RD-013.1
@controller(EMR Document Processing) provides too late @action(list_new_documents) because the time interval is not short enought or the EMR system is not configured to list new documents automatically.

- Mitigated by @req(CC-054)
- New @req(CC-112)

[H-04]

UCA: UCA-search_document-not_provided

Scenario: LS-RD-014.1
@controller(EMR Document Processing) does not provide @action(search_document) for a missed document (due to a fail in@action(get_document)) because the EMR System does not keep track of documents that could not be retrieved.

- New @req(CC-115)

[H-04]

UCA: UCA-get_document-not_provided

Scenario: LS-RD-015.1
@controller(EMR Document Processing) does not provide @action(get_document) for a new document because the EMR system is not configured to get documents automatically.

- Mitigated by @req(CC-055)
- New @req(CC-113)

[H-04]

UCA: UCA-get_document-provided_too_late

Scenario: LS-RD-016.1
@controller(EMR Document Processing) provides too late @action(get_document) for a new document because the time interval is not short enought or the EMR system is not configured to get documents automatically.

- Mitigated by @req(CC-055)
- New @req(CC-113)

[H-04]

Control Action: EMR Document Processing - get_document

Scenario: LS-RD-017.1
@controller(EMR Document Processing) provides @action(get_document) but does not notify the users about new documents.

- Mitigated by @req(CC-018)

[H-04]

UCA: UCA-get_document-stopped_too_soon

Scenario: LS-RD-018.1
@controller(EMR Document Processing) stops too soon providing @action(get_document) for a new document due to a connection error with the CDX system.

- Mitigated by @req(CC-054)
- New @req(CC-115)

[H-04]

UCA: UCA-get_document-provided_repeatedly

Scenario: LS-RD-019.1
@controller(EMR Document Processing)provides @action(get_document)repeatedly for a document already received, and the EM system saves the downloads as new documents, or as updates of the already received document.

- New @req(CC-116)

[H-05]

UCA: UCA-retain_replaced_document-not_provided

Scenario: LS-RD-020.1
@controller(EMR Document Processing) does not provide @action(retain_replaced_document) after @action(replace_document) for an updated document, instead the EMR system deletes previous versions of the update document.

- Mitigated by @req(CC-015) and @req(CC-019)

[H-04]

UCA: UCA-replace_document-not_provided

Scenario: LS-RD-021.1
@controller(EMR Document Processing) does not provide @action(replace_document) for an updated document, instead the EMR system treats the documents versions as different documents.

- Mitigated by @req(CC-014)

[H-05]

UCA: UCA-replace_document-provided_out_of_sequence

Scenario: LS-RD-022.1
@controller(EMR Document Processing) provides @action(replace_document) out of sequence for two or more versions of the same document, saving them in wrong order, because the document arrived out of order and the version number is not considered.

- Mitigated by @req(CC-014)

[H-04]

UCA: UCA-auto_assign_provider-not_provided

Scenario: LS-RD-023.1
@controller(EMR Document Processing) does not provide @action(auto_assign_provider) for received document because the provider is not registered within the EMR system or the registry is incomplete (missing identifier).

- New @req(CC-114)

[H-04]

UCA: UCA-auto_match_patient-not_provided

Scenario: LS-RD-24.1
@controller(EMR Document Processing) does not provide @action(auto_match_patient) for a received document because the patient is not registered within the EMR system or the registry is incomplete.

- Mitigated by @req(CC-008), @req(CC-009) and @req(CC-010)

[H-04]

UCA: UCA-auto_match_patient-provided_for_wrong_patient

Scenario: LS-RD-025.1
@controller(EMR Document Processing) provides @action(auto_match_patient) for the wrong patient because the registered patient information is incorrect or incomplete.

- Mitigated by @req(CC-007)
- New @req(CC-117)

[H-04]

UCA: UCA-store_attachment-not_provided

Scenario: LS-RD-026.1
@controller(EMR Document Processing) does not provide @action(store_attachment) for a received document because the EMR system does not extract the attachments from the documents.

- Mitigated by @req(CC-020) and @req(CC-022)

[H-03]

UCA: UCA-store_attachment-provided_for_wrong_patient

Scenario: LS-RD-027.1
@controller(EMR Document Processing) provides @action(store_attachment) for the wrong patient because the matched patient is not correct.

- New @req(CC-117)

[H-03]

UCA: UCA-link_document-not_provided_for_report

Scenario: LS-RD-028.1
@controller(EMR Document Processing) does not provide @action(link_document) for a received document (report) because the EMR system does not link related documents.

- Mitigated by @req(CC-043)

[H-06]
Scenario: LS-RD-028.2
@controller(EMR Document Processing)does not provide@action(link_document)for a received document (report) because the related document (order) was deleted or not saved.

- New @req(CC-119)

[H-06]

Control Action: EMR Document Processing - list_new_documents

Scenario: LS-RD-029.1
@controller(EMR Document Processing) provides @action(list_new_documents), but the list of new documents is not returned due to a connection issue or error on the CDX system.

- Mitigated by @req(CC-058)
- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-04]

Control Action: EMR Document Processing - search_document

Scenario: LS-RD-029.1
@controller(EMR Document Processing) provides @action(search_document), but the list of documents is not returned due to a connection issue or error on the CDX system.

- Mitigated by @req(CC-058)
- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-04]

Control Action: EMR Document Processing - get_document

Scenario: LS-RD-030.1
@controller(EMR Document Processing) provides @action(get_document), but the document is not returned due to a connection issue or error on the CDX system.

- Mitigated by @req(CC-058)
- New @req(CC-115)
- Suggestion: System or users should decide if they will try it again when the connection is re-established.

[H-04]

Appendix D

New Controller Constraints

6_New Constraints

model cdx.interoperability

Req CC-101 : Preview document using the approved document viewer before submission.

kind: controller constraint - associated control actions:
uca: UCA-create_document-provided_with_wrong_content
uca: UCA-update_document-provided_with_wrong_content

Req CC-102 : Document content is included in documents being created.

kind: controller constraint - associated control actions:
uca: UCA-create_document-provided_with_wrong_content

Req CC-103 : When linking a document in a clinical workflow, a brief summary of the linked document is displayed.

kind: controller constraint - associated control actions:
uca: UCA-create_document-provided_for_wrong_order

Req CC-104 : Multiple submission of the same document is prevented.

kind: controller constraint - associated control actions:
uca: UCA-create_document-provided_repeatedly
uca: UCA-update_document-provided_repeatedly

Req CC-105 : After starting the composition of the document, confirm to abort the document submission.

kind: controller constraint - associated control actions:
uca: UCA-create_document-stopped_too_soon
uca: UCA-update_document-stopped_too_soon
uca: UCA-cancel_document_stopped_too_soon

Req CC-106 : Errors in the submission of the document are indicated.

kind: controller constraint - associated control actions:
uca: UCA-create_document-stopped_too_soon
uca: UCA-update_document-stopped_too_soon
uca: UCA-cancel_document_stopped_too_soon

Req CC-107 : Users are notified if submitted documents are not timely received or any error happened during the delivery .

kind: controller constraint - associated control actions:
controller: EMR Document Entry **- action/feedback:** get_document_status

---

Req CC-108 : When updating or cancelling a document, the previous content is rendered.

kind: controller constraint - associated control actions:
uca: UCA-update_document-provided_for_wrong_document
uca: UCA-cancel_document-provided_for_wrong_document

Req CC-109 : Only the latest version of a document can be updated.

kind: controller constraint - associated control actions:
uca: UCA-update_document-provided_for_wrong_document
uca: UCA-update_document-provided_for_wrong_document

Req CC-110 : Record target (patient) of a sent document cannot be updated.

kind: controller constraint - associated control actions:
uca: UCA-update_document-provided_with_wrong_target

Req CC-111 : When updating a document, recipients can only be added, not removed.

kind: controller constraint - associated control actions:
uca: UCA-update_document-provided_with_wrong_recipient

Req CC-112 : Listing of new documents is performed automatically at reasonable time intervals.

kind: controller constraint - associated control actions:
uca: UCA-list_new_documents-not_provided
uca: UCA-list_new_documents-provided_too_late

Req CC-113 : Get documents is performed automatically at reasonable time intervals.

kind: controller constraint - associated control actions:
uca: UCA-get_document-provided_too_late

Req CC-114 : EMR system keeps the information and identification of the registered providers of the clinic up to date.

kind: controller constraint - associated control actions:
uca: UCA-auto_assign_provider-not_provided

Req CC-115 : Receiving system is responsible for keeping track of problems in getting documents.

kind: controller constraint - associated control actions:
uca: UCA-get_document-stopped_too_soon

Req CC-116 : Receiving system stores a received document only once; even the document is retrieved multiple times.

kind: controller constraint - associated control actions:
uca: UCA-get_document-provided_repeatedly

Req CC-117 : Receiving system allows a document to be manually matched to another patient in case it is associated to the wrong patient.

kind: controller constraint - associated control actions:
uca: UCA-store_attachment-provided_for_wrong_patient

Req CC-118 : Users are notified when new versions of documents are received.

kind: controller constraint - associated control actions:
controller: EMR Document Processing **- action/feedback:** document_updated

---

Req CC-119 : Receiving system indicates when documents have relationships (events, orders, related docs, doc version) not fulfilled.

kind: controller constraint - associated control actions:
uca: UCA-link_document-not_provided_for_report
