Institute of Software Engineering
University of Stuttgart
Universitätsstraße 38
70569 Stuttgart
Germany

Bachelor’s Thesis

Applicability Analysis: Elicitation of Privacy Risks through STPA(-Priv) in a Selected IoT-Scenario

Frederik Riedel

Course of Study: Software Engineering

Examiner: Prof. Dr. Stefan Wagner

Supervisor: Kai Mindermann

Commenced: October 5, 2016

Completed: April 6, 2017

CR-Classification: K.4.1, D.2.1

Abstract

Context

This bachelor's thesis discusses the usage of *System-Theoretic Process Analysis (STPA)* for *privacy engineering*. STPA has been developed for safety engineering originally. I show how this methodology can be applied to privacy risk analysis by using the extension *STPA-Priv*. I explain why privacy is important and why privacy risk analysis can help improve systems regarding privacy.

Objective

The goal is to apply the privacy extension of STPA to a real-world Internet of Things scenario to determine the applicability and possible problems with this methodology.

Method

STPA considers safety a system property. I think that privacy is a system property as well and therefore STPA can be applied to privacy risk analysis. Most changes from STPA to STPA-Priv have been made in its terminology, the process itself remains the same. This brings many of the advantages of systems theory to the field of privacy engineering, such as the top-down nature of STPA that helps handle complex socio-technical systems.

Results

I found out that STPA-Priv is a good approach to elicit privacy risks and requirements. I was able to elicit many privacy risks from our scenario using STPA-Priv which shows that the methodology works in general.

Conclusions

After all, I can recommend using STPA-Priv to evaluate projects for privacy risks. Nevertheless, there are still changes and improvements necessary. However, the overall methodology would not be affected by those changes. STPA-Priv is very straight-forward for people that are already familiar with STPA.

Zusammenfassung

Kontext

Diese Bachelorarbeit diskutiert die Verwendung *systemtheoretischer Prozessanalyse (STPA)* zur Analyse von *Privatsphärerisiken* in *Systemen*, die sich aus Hardware- und Softwarekomponenten zusammensetzen. Aber auch externe Faktoren sollen berücksichtigt werden. STPA wurde ursprünglich für die Sicherheitstechnik entwickelt. Ich zeige, wie STPA durch die Erweiterung STPA-Priv auch für die Privatsphärerisikoanalyse angewendet werden kann. Ich erkläre, warum Privatsphäre wichtig ist und wie STPA-Priv dabei helfen kann, Privatsphäre und Datenschutz in Systemen zu verbessern.

Zielsetzung

Das Ziel ist es, die Erweiterung von STPA, *STPA-Priv*, auf ein echtes *Internet of Things* Szenario anzuwenden. Die Anwendung zeigt, inwiefern dieses Verfahren tatsächlich zur Erhebung von Privatsphärerisiken anwendbar ist. Außerdem werden mögliche Probleme und Schwierigkeiten aufgedeckt.

Methode

Für die Anwendung von STPA geht man davon aus, dass *Sicherheit* eine Systemeigenschaft ist. Ich gehe davon aus, dass Privatsphäre auch eine Systemeigenschaft ist und dass STPA daher für die Analyse der Datenschutzrisiken angewendet werden kann. Die meisten Änderungen von STPA-Priv gegenüber STPA bestehen lediglich in der Abänderung der Terminologie, der Prozess selbst bleibt der gleiche. Somit können die Vorteile der Systemtheorie auch für die Privatsphärerisikoanalyse genutzt werden, beispielsweise der Top-down Ansatz von STPA, der dabei hilft komplexe soziotechnische Systeme zu handhaben.

Ergebnisse

Ich habe herausgefunden, dass STPA-Priv ein guter Ansatz ist, um Privatsphärerisiken und Anforderungen für Datenschutz zu ermitteln. Ich konnte viele Datenschutzrisiken aus meinem Szenario durch STPA-Priv herausfinden, was zeigt, dass die Methodik im Allgemeinen funktioniert.

Schlussfolgerungen

Nachdem ich STPA-Priv selber auf ein Szenario angewandt habe, kann ich die Verwendung von STPA-Priv zur Privatsphärerisikoanalyse empfehlen. Jedoch sind noch Änderungen und Verbesserungen notwendig. Die grundlegende Struktur von STPA-Priv wird allerdings nicht durch diese Änderungen beeinflusst. Wer bereits mit STPA vertraut ist, wird sich in STPA-Priv schnell zurecht finden.

Contents

1. Introduction

1.1. Motivation

The importance of privacy is increasingly relevant for organizations and individuals in a connected world. Complex socio-technical software systems offer personalized services, personal assistants, and cloud services. Collecting and processing personal information is essential for those services. Governments, social networks, search engines, insurance companies, banks, and other organizations and institutions already collect and store massive amounts of privacy-sensitive data. We now live in an age where digital storage options have become so cheap that recording and accumulating all generated and collected data is not a problem anymore [1]. Companies can investigate intentions, interests, needs, desires, and fears of each user by evaluating these data sets. This data enables them to specifically target users by providing manipulative information [2]. More and more devices are connected to the Internet — *ubiquitous computing* really becomes ubiquitous in every-day life: The *GSM Association* predicts a number of 24 billion inter-connected devices by 2020 [3]; including toothbrushes, refrigerators, television devices, toasters with different sensors like cameras, microphones, GPS (global positioning system) sensors, motion, temperature and light sensors. Privacy-related risks are often not considered by companies when recording or transferring this information to other participants or companies. They often stay unconsidered intentionally — just to be able to collect more private information [4]. However, “Privacy is a fundamental part of human dignity. It is the human right to refuse interference by others in one’s life.”, as the European Commission states [5]. Privacy risk analysis could help prevent unregulated collection and processing of privacy sensitive information. The general idea is to utilize *systems theory* in order to elicit privacy risks from systems. Systems theory has many advantages compared to traditional analysis techniques, especially because it treats the system as a whole and does not focus on issues on sub-module level. *System-theoretic process analysis* approaches consider *safety* and *security* system properties. I expect that privacy can be considered a system property as well, this is why system-theoretic approaches could also be applied to privacy. System-theoretic process analysis (STPA) has been developed to analyze systems for safety-risks and has been extended for the use with security as well. I expect that *STPA* can be used for privacy-risk analysis as well. The recently proposed *STPA-Priv* [6] methodology for privacy risk analysis is still new so there is almost no literature available on this topic. I want to analyze the feasibility and applicability of STPA-Priv for privacy analysis in this thesis.

1.2. Goals

System-theoretic process analysis (STPA) has already been successfully applied to safety (STPA) and security (STPA-Sec) analysis. Privacy has not been considered as target for the STPA methodology so far — but with the emerging need for privacy-aware systems this approach becomes interesting. The goal is to be able to support engineers to plan and execute privacy-aware systems using *STPA-Priv* [6]. As it is a new approach there are not many sources that examine the applicability of STPA-Priv including examples, tests, advantages and disadvantages. I want to explore this by applying the method to a complex scenario. In this thesis I want to find out how STPA-Priv performs in a real-world scenario and what changes might be necessary to make it better and more easily applicable. I also take a look at tool-support for STPA and how this could help for STPA-Priv as well.

1.3. Thesis Structure

This bachelor's thesis is structured in four main chapters: The first one, *Foundations*, explains basic concepts that are necessary for STPA-Priv, including *systems theory* itself, and *system-theoretic process analysis (STPA)* is described using a detailed sample scenario. The security extension *STPA-Sec* is explained as well. The software *XSTAMPP* which is optimized to support safety engineers is introduced and the importance of eHealth is underlined by giving an example scenario that is used later on for STPA-Priv analysis. This chapter also includes important basics of *Privacy and Privacy Risk Models*. The next chapter lists *Related Work and Existing Privacy Analysis Techniques* and shows the advantages and disadvantages of each methodology. The following chapter describes how STPA can be *extended for privacy engineering*, and the last chapter shows how *STPA-Priv can be used in the real-world eHealth scenario*. In the end, I give a conclusion which states how applicable STPA-Priv already is and which changes and research are still necessary.

Parts of this thesis have been submitted as a paper that I wrote together with Kai Mindermann, Asim Abdulkhaleq, Christoph Stach and Stefan Wagner. The paper is titled "*Investigating the Applicability of STPA-Priv for Privacy Engineering*". If accepted, this paper will be published in the journal *Proceedings on Privacy Enhancing Technologies* by *De Gruyter Open* in June 2017.

2. Foundations

2.1. Systems Theory

2.1.1. Systems

When talking about systems theory it is important to understand what a system itself is. A system is "a regularly interacting or interdependent group of items forming a unified whole" [7]. When it comes to privacy I am mostly speaking about software-systems consisting of submodules. However, hardware-components are also part of the system and need to be considered. After all, a system contains all modules, components and sub-systems that are necessary and essential for running the whole system.

Systems theory started developing during the 1930s and 1940s in response to the limitations of traditional analysis techniques [8]. New theoretical, epistemic and mathematical problems in systems with increasing amounts of complexity required new analysis techniques [9]. Since then different approaches have been developed to study the nature of complex systems of biologic organisms, physicochemical systems, psychic systems, social systems and machines such as computers with its software-systems [8, 10].

2.1.2. Traditional Techniques

Traditional analysis techniques such as *analytic reduction* try to break down systems into smaller parts to be examined separately — also known as *divide and conquer* [11]. Systems are examined from two different perspectives: Time-independent characteristics are separated into their hardware and software subsystems, whereas time-sensitive activities are divided into isolated events [11]. This approach anticipates that each component or event works free from external control and constraints to other subsystems or events [11]. The idea of these approaches is that isolated events and isolated subsystems operate independently from other. However, system behavior in complex socio-technical systems cannot always be reduced to single components or events. This is where systems theory improves analysis. The focus is not on single components or events but on the system as a whole.

2.1.3. Advantages of Systems Theory

In contrast to analytic reduction approaches systems theory considers the system as a whole. It is not divided into components or subsystems nor limited to the technical aspects of the system. The related environment such as humans interacting with the system are also considered [11]. This is especially important in today's complex socio-technical systems, where humans and other environmental conditions interact with the system.

2.1.4. Disadvantages of Systems Theory

When looking at the system as a whole, it is often criticized that the high-level system structure does not recognize details on lower levels that might be relevant for the behavior of the system. This could lead to wrong assumptions when dealing with the high-level architecture. It is hard to have a complete overview over complex systems.

2.1.5. Hierarchy and Abstraction

Systems theory follows the rules of hierarchy and levels of organization. Systems are not only organized by their submodule's competences or determination like in traditional approaches, but certain levels of abstraction are created to be able to manage complex tasks. Higher levels of abstraction control lower levels, whereas lover levels provide feedback to higher levels.

2.2. System-Theoretic Process Analysis (STPA)

Many methodologies and tools can be used for hazard analysis [12]. *Fault Tree Analysis* (FTA) [13], *Failure Mode and Effect Analysis* (FMEA) [14] and *Hazard and Operability Analysis* (HAZOP) [15] are among the most commonly used approaches. Those *traditional hazard analysis techniques* mostly use sequential or epidemiological chains of causality, assuming that failures in a system can be traced back to single component failure [8, 16].

2.2.1. Systems-Theoretic Accident Modeling and Processes (STAMP)

STPA is based on the *Systems-Theoretic Accident Modeling and Processes* (STAMP) which has been developed by Leveson [8]. The STAMP approach assumes that accidents result from insufficient enforcement of safety constraints in system design, development, and operation. A system is considered safe when no safety constraints are violated. The systems-theoretic approach treats safety as a control problem instead of isolated component failure. Systems theory allows to consider more complex socio-technical systems including human interaction, system dependencies, or environmental influences. STAMP considers failures to be a result of safety constraint violations in component interactions [8].

2.2.2. From STAMP to System-Theoretic Process Analysis (STPA)

STPA is an analysis technique that has been developed to support the analysis of safety risks within systems, based on STAMP. The most important aspect of STPA is its systems theoretic approach for process analysis. This allows to consider indirect relationships between events and social components surrounding the system as well as internal dependencies. STPA is a top-down procedure in contrast to traditional bottom-up causality models which only focus on single component failures. Accidents can be the outcome of complex system component interactions which might be not visible on the component level. This includes human mistakes, management and organizational errors as well as hardware and software failures violating system safety constraints. Indeed, system accidents can occur without a single component failing [17]. STPA tries to solve this issue and can therefore be useful for drafting systems and systems verification. For each accident different hazards can be defined which can lead to the accident — after all, occurrences of those hazards should be prevented within the system. This is done by defining unsafe control actions and analyzing the control structure to find all possible causes for these hazards.

The methodology of STPA consists of three basic steps [8]:

1) The fundamental analysis starts with listing losses that should not occur in the system. *Losses* cause damage to humans, the environment, or the system itself. Each loss can be assigned to one or more *hazardous system states* in which the system could result in a loss. It is important to differentiate between losses and hazards: Losses are not under the system's control — they are accidents that can happen under certain worst-case conditions. These worst-case conditions are called hazardous system states. And they are, in contrast to losses, under the system's control. The idea is to prevent hazardous system states to prevent losses. This is where safety constraints come into place: *Safety constraints* ensure that the system never results in a hazardous state. They are created by negating hazards and writing them down as stand-alone

statement. Finally, one needs to create a *control structure diagram* of the system — such a diagram should already exist for existing systems, but systems that are currently drafted might require creating a new control structure diagram.

2) *STPA Step 1:* This step is about identifying unsafe control actions. Unsafe control actions are those control actions that could violate safety constraints from 1). Each unsafe control action is classified as *not providing causes hazard*, *providing causes hazard*, *incorrect timing or wrong order causes hazard* or *stopped too soon or applied too long causes hazard*.

3) *STPA Step 2:* The last step of STPA is to generate causal scenarios for each unsafe control action. This is important to understand *how* a hazardous system state could occur in addition to the results of 2) that show *why* a hazardous system state could occur.

This concludes the methodology of STPA. The risk management can then evaluate those risks and can take countermeasures.

2.2.3. Control Structure Diagram

Control structure diagrams are high-level charts which visualize dependencies between participants and components within a system. It reflects the hierarchical structure and layers of abstraction within the system. Interacting components are connected with an annotated arrow. This arrow represents a control action or a feedback loop and can have different attributes which contain information about the relationship and exchanged data types.

Control structure diagrams visualize controllers, actuators, processes and sensors in so-called feedback loops. Controllers send control actions to actuators and receive feedback from sensors on the process, as illustrated in Figure 1.

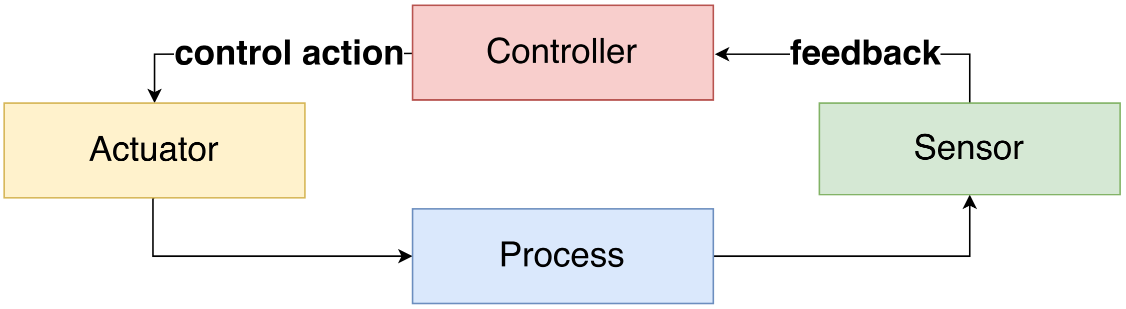


Figure 1: A simple control structure diagram featuring one control loop. The controller sends a control action to the actuator, and receives feedback on the process from sensors.

2.2.4. Example Problem: Batch Chemical Reactor

This example shows how a system can fail and result in a hazard despite every component is working as expected. The batch chemical reactor has been used for pharmaceutical productions in the UK [8].

The simplified control structure of the batch chemical reactor can be seen in Figure 1. The main component of this system is the reactor in which different chemicals and catalysts react to produce different chemicals. A valve can be opened to start filling a catalyst into the reactor. The cooling system prevents toxic vapors from venting and instead makes them reflux into the reactor. The cooling system can be controlled by a valve as well which then enables or disables the functionality of the condenser.

Everything is controlled by a software-system. After the software-system opens the valve for catalysts to flow into the reactor the reaction will start and get hot. This is why the software-system is designed to activate the cooling system afterwards to enable the reflux of chemicals.

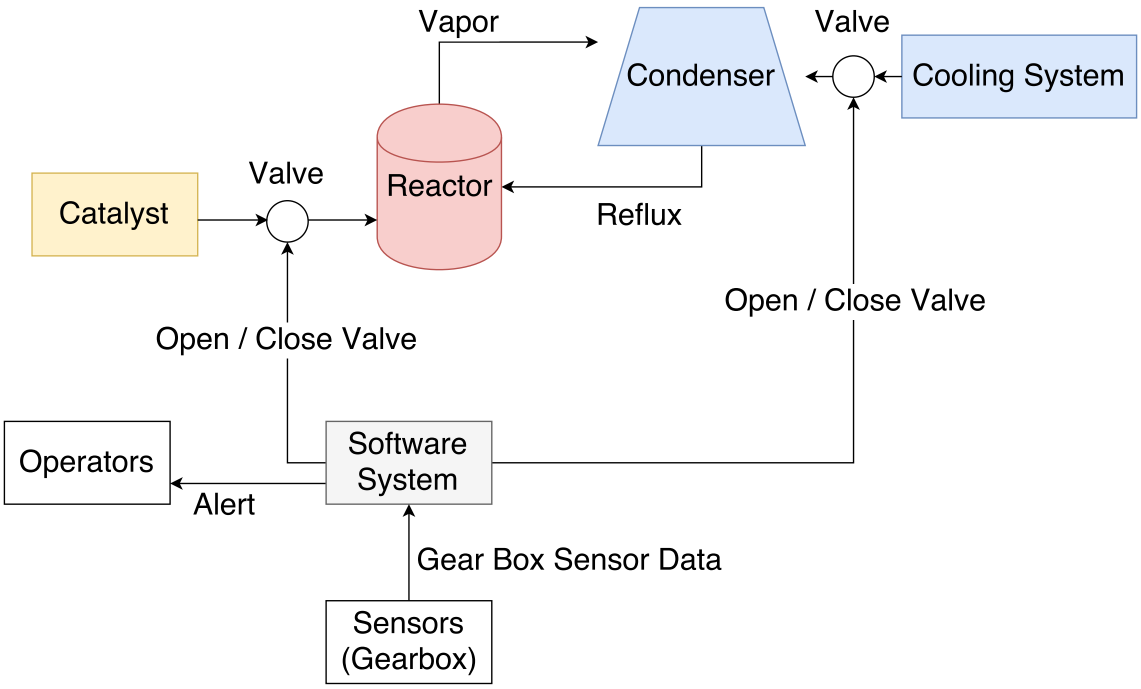


Figure 2: Model of a batch chemical reactor. The reaction itself is happening in the reactor in the center. Different valves control the conflux of the catalyst and can enable the cooling system in the condenser. All valves are controlled by a software system.

Different sensors, for instance oil-sensors in the gear-box, can provide feedback to the software-system. The software then decides what to do next. Whenever sensors report problematic data to the computer, the computer should stop controlling the system and instead send an alert to the operators to eliminate the problem. This procedure has been defined by the systems engineering team.

After all, the following scenario can occur: The computer opens the catalyst valve to start the reaction. Shortly after that some sensors report a problem with the oil in the gear — this stops further actions of the computer and alerts the operators to take a look at the system and identify the problem. While the operators are refilling the oil in the gear, the reactor already gets hot and produces vapor which is released to the condenser. However, the computer was not able to start the cooling system so far — because system requirements tell the computer that it needs to wait for the operators to continue the process. This results in a vent of toxic chemicals.

This example shows why STPA is better than traditional hazard analysis strategies: every component did what it was supposed to do, but still a hazard occurred.

Traditional hazard analysis focuses on each single component to work without problems, but does not consider dependencies between sub-systems. In this scenario, STPA could have prevented this hazard [8, 18].

The following sections describe the general approach of STPA and how it works using the chemical batch reactor as an example. Chapter 4 then extends STPA to be used for privacy analysis.

2.2.5. Defining Accidents and Hazards

The first step of STPA is all about finding possible safety-related accidents (losses) and system-states that can lead to such accidents. These system states together with their environmental conditions are called hazards. Hazards are system states that are under the system's control, whereas accidents themselves are not controllable. The idea is to define constraints around these hazards to ensure that these cases do not occur and therefore do not lead to accidents [8].

Table 1 shows examples of accidents and hazards: Hazards are system-states that can lead to accidents, for example the release of toxic chemicals can lead to the expose of people to these chemicals. I want to prevent these accidents, therefore I need to prevent all hazards that can lead to those accidents. Workers could also get roped into the gear, while refilling the oil. In this case the hazard is that the gear is not stopped when oil is being refilled. Another accident that can happen is that workers are exposed to toxic chemicals. This can caused by a hazardous system state in which the reactor has not been cleaned correctly before being opened.

Accident or Loss	System Hazard
People are exposed to toxic chemicals.	Water cooling system not active while catalyst in reactor.
Workers get roped into the gear.	Gear is not stopped when oil is being refilled.
Workers are exposed to toxic chemicals.	Reactor is not cleaned correctly before being opened.

Table 1: Examples of Accidents and Hazards in the STPA Model using the batch chemical reactor scenario [6, 8, 18, 19].

2.2.6. Derive Safety Constraints from Identified System Hazards

The idea of safety constraints is to have list of requirements for the system that must be ensured in order to prevent hazardous system states. These safety constraints must be valid while the system is running to prevent hazardous system states. Safety constraints are generated from the hazard list. The results can be seen in Table 2.

System Hazard	Safety Constraint
Water cooling system not active while catalyst in reactor.	The water cooling system must be active when the catalyst is filled into the reactor.
Gear is not stopped when oil is being refilled.	The gear must be stopped when oil is being refilled.
Reactor is not cleaned before being opened.	The reactor must be cleaned before being opened.

Table 2: Derived safety constraints from system hazards.

2.2.7. Control Structure Diagram

Control structure diagrams are high-level charts which visualize dependencies between participants and components within a system.

It is important to pay attention to control and feedback actions between components. Different components have different scopes. Some components control other components, whereas other components receive commands and provide feedback on actions and sensor data. The control structure diagram must contain information about controllers and a list of commands and possible feedback that is being sent between two components. All these aspects are then arranged in a control structure diagram.

The software system is the central controller of the system. It is able to start and stop the catalyst supply and to start and stop the cooling system. Sensors within the gearbox report failures and provide feedback. The software system can receive input commands from operators and provides feedback to them by sending alerts.

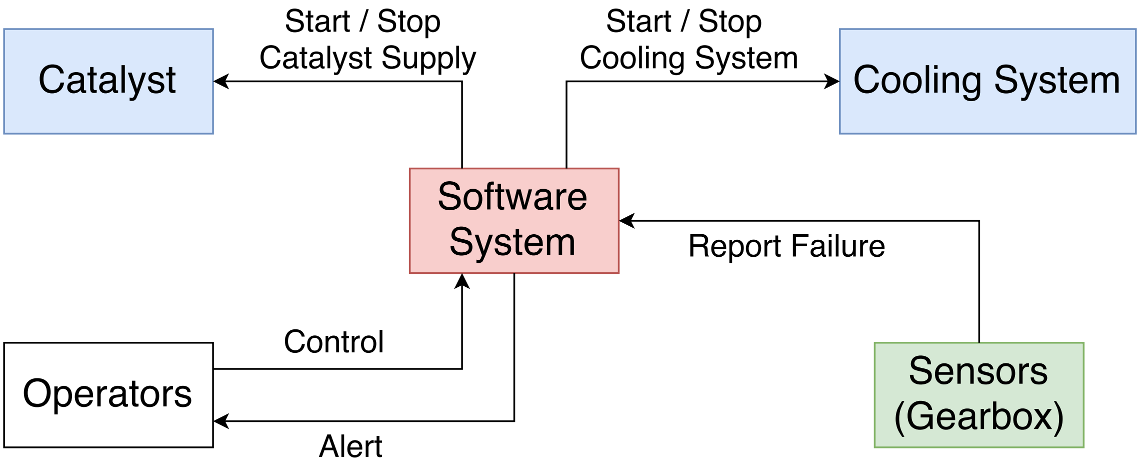


Figure 3: Control structure diagram of the batch chemical reactor scenario.

2.2.8. Step 1: Identify Unsafe Control Actions (UCAs)

Unsafe control actions are control actions that can violate safety constraints when executed. The goal of this step is to find control actions that violate safety constraints and therefore can lead to hazardous system states. In the end, these unsafe control actions are the flaw of my system and need to be prevented by engineers.

Unsafe control actions can be generated by filling out the unsafe control action table. This table includes all control actions from the control structure diagram. The table provides different types for each control action which can violate safety constraints: not providing causes hazard, providing causes hazard, incorrect timing or wrong order causes hazard, stopped too soon or applied too long causes hazard, as seen in Table 1. Control actions in combination with these categories lead to unsafe control actions. All hazardous system states are underlined in Table 3.

Unsafe control actions can be caused by a mis-communication between components. Withholding information, providing information or providing information with bad timing can lead to hazardous system states [6]. This is where the four categories of

Control Action	Violated safety constraint	Not providing causes hazard	Providing causes hazard	Incorrect timing / wrong order causes hazard	Stopped too soon / applied too long causes hazard
Start Catalyst Supply	The water cooling system must be active when the catalyst is filled into the reactor.		Starting when cooling system offline	Starting more than X seconds before cooling system started	
Stop Catalyst Supply	The water cooling system must be active when the catalyst is filled into the reactor.	Not stopping when cooling system offline		Stopping catalyst after cooling system has been stopped	
Start Cooling System	The water cooling system must be active when the catalyst is filled into the reactor.	Not started when catalyst supply started		Starting more than X seconds after catalyst supply started	
Stop Cooling System	The water cooling system must be active when the catalyst is filled into the reactor.		Stopping while catalyst supply is active	Stopping before catalyst supply stopped	Stopping before catalyst supply stopped

Table 3: Examples of control actions and unsafe control actions using the batch chemical reactor scenario [6, 8, 18, 19]. Hazardous system states are underlined.

2.2.9. Step 2: Generating Causal Scenarios

The last step of STPA concludes causal factors for unsafe control actions. The goal is to find scenarios in the system that can lead to unsafe system states. This is not limited to simple components, but can occur in conjunction with components and control actions within the whole system.

In my scenario different problems can be determined: After analyzing the unsafe control action table one can see that the cooling system must always enabled before the catalyst supply is started. It is also important to receive feedback from all sub-systems, when failures occur. For example, the cooling system failing for some reason requires stopping the catalyst supply as well.

2.3. STPA-Sec: STPA for Security

Young and Leveson [20] developed an extension of STPA for security analysis. This extension is called STPA-Sec (STPA for security). The general procedure remains the same which makes it consistent for users that have already experience with STPA. However, some terminology needed to be changed in order to make the approach coherent for security: Hazards are called vulnerabilities, unsafe control actions are called insecure control actions. Additionally there is a change when generating causal scenarios: Whereas safety analysts try to find scenarios in which unintentional actions can lead to losses, security analysts also need to find scenarios in which intentional actions can lead to losses caused by malevolent actors [20]. After all, the three steps of STPA-Sec are: 1) Identify losses and vulnerabilities, create the control structure diagram; 2) identify insecure control actions; 3) identify causal scenarios for insecure control actions considering intentional actions.

2.4. eXtensible STAMP platform (XSTAMPP)

XSTAMPP[1] (eXtensible STAMP platform) is a software platform that helps engineers to implement the STAMP procedure and is available under the open-source license “Eclipse public license” [21]. The tool has been published and is maintained by Balzer, Abdulkhaleq and Wagner [22]. XSTAMPP is built on top of Eclipse using the Eclipse Plugin Development Environment and Rich Client Platform. Many plugins are available for the current version XSTAMPP 2.1.1. These plugins offer support for STPA for safety engineering, STPA-Sec for security engineering, and STPA-Priv for privacy engineering. I will be using XSTAMPP with its STPA-Priv plugin in my scenario to support the process of identifying losses, hazards, safety constraints or their appropriate correspondents in privacy engineering.

[1] http://www.xstampp.de

2.5. Internet of Things (IoT) and Electronic Health (eHealth)

The *internet of things* has already become part of the everyday life in many households and companies. Smart devices that control and capture information are widely spread in cars, homes and offices. This trend will continue in the next years, since there will be more than 24 billion connected devices by the year of 2020 [3]. Smartphones are a great example of how technology spreads and becomes part of the everyday life. In his publication “Generation Smartphone” [23] Siewiorek describes how smartphones can help humans handle tasks, coach them or monitor them with existing technology and future technologies. This includes health-related issues ranging from detecting tired drivers in cars to capturing essential vital functions [23]. Smartphones can already monitor many factors of our life, including sensors and algorithms such as GPS location, emotions (audio analysis, face recognition, heart rate) or posture (accelerometer) [24], just to name a few. Smartphones, and smart devices in general, are already part of the ubiquitous computing *internet of things* network today. Their influence will be growing in the future. The amount of data they can monitor is stunning and holds advantages for health and living conditions as well. Smart devices will be able to perform periodic screenings at home without the presence of a physician. Chronic diseases such as diabetes mellitus challenge the healthcare system, facing high treatment costs and overstrained physicians. Smart devices can help to overcome these problems by increasing the number of preventive medical checkups, allowing a comprehensive monitoring of patients, and lowering the costs.

Serious games are a use case of eHealth technology. They can help integrate eHealth treatments into the daily routine of young patients. Especially children are able to handle their disease much better, because the game reminds and motivates them to comply with the therapy [25]. However, users of such applications are not the only stakeholders that can profit from eHealth technology. In his publication [25] Knöll describes how traditional eHealth data of smart devices could be enriched with additional contextual data. This helps researchers to find correlations between the health state and the person’s environment. As an example, *unhealthy* places in a city could be identified by urban planners [25].

Internet of things and electronic health approaches are very controversial when it comes to data collection and sharing. On the one hand the society could profit from huge accumulated data sets, that could help to identify unhealthy places, as suggested by Knöll [25]. On the other hand many patients are afraid to provide their sensitive health data to third parties. Security in general and privacy in particular are the biggest concerns [26, 27]. Their awareness for privacy and security is great at first. However, refusing to use a service completely does not solve the privacy concerns in the first place. I think that a strategic use of a privacy risk analysis methodology such as STPA-Priv could help make such solutions safe and secure regarding privacy and, in the end, help make them socially acceptable.

2.6. Privacy and Privacy Risk Models

Privacy often overlaps with fields of security but both live within their own scope and have a right to exist. Security has been described by Saltzer and Schroeder in their 1975-released paper as “mechanisms and techniques that control who may use or modify the computer or the information stored in it” [28] whereas privacy is described as “the ability of an individual (or organization) to decide whether, when and to whom personal (or organizational) information is released” [28]. However, their descriptions lack the consideration of computer networks, such as the internet. In general, security deals with defending actively attacking enemies who want to access information. This can happen remotely or while having hardware access. Privacy ensures that information that has been revealed to specific entities in moderation and under certain conditions is not disclosed unintentionally to prevent inconvenient or adverse consequences.

Privacy covers a wide range of multi-dimensional issues [29]. The perception of privacy can differ among cultures, countries and institutions depending on their values and interests. The European Commission published a list of possibilities that can be applied to data when it comes to processing. This includes, but is not limited to: collection, organization, combination or disclosure of data [30]. The full list can be seen in Figure 4.

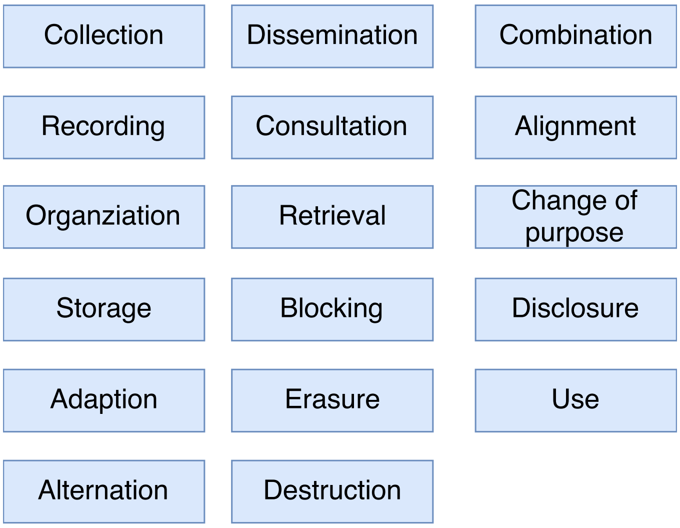


Figure 4: An overview of possibilities to process data, according to the European Commission [30].

Privacy risk models draw a framework under which a common understanding of the wide field of privacy is possible. They help to understand which aspects of a system need to be considered when analyzing privacy risks. Hereinafter, I present two examples of privacy risk models. The first one, *Calo's privacy harms*, [31] is a more general approach that still covers the most important privacy harms, whereas the *LINDDUN privacy threat catalog* [32] covers privacy in depth and very detailed.

2.6.1. Calo's subjective/objective privacy harms

Calo's privacy harms [31] cover a rather general framework for privacy harms. They are split into two groups: Subjective privacy harms cover the perception of unwanted surveillance, objective privacy harms cover the usage of private information. The author argues that most privacy harms can be categorized in either one of those categories. Subjective privacy harms cover incidents such as observation, whereas objective harms cover incidents such as identity theft or using a suspect's blood alcohol level as incident against them.

2.6.2. LINDDUN Privacy Threat Framework

The *LINDDUN privacy threat tree*, as described by Wuyts, Scandariato and Joosen [32], is very detailed and covers a wide range of issues. This threat tree is part of the comprehensive LINDDUN methodology which includes a data flow analysis as well. The threat framework drawn by LINDDUN includes threat trees on the following issues:

- *Linkability* covers activity where data collected from different users or groups is being linked, but also includes different (anonymous) data sets that are linked to the same person.

- *Identifiability* covers scenarios where subjects and anonymous data sets can be identified.

- *Non-repudiation* is the case when a user of a system is not able to deny a claim because the system has evidence that can prove the claim.

- *Detectability* covers activity where a system can ensure that a data set of interest exists — but might not have access to it. However, the knowledge that a specific data set exists can already reveal certain conclusions.

- *Information Disclosure* occurs when information is revealed unintentionally. This heavily depends on security.

- *Unawareness* describes situations in which the user is not aware of consequences that can occur when sharing their information or that information is shared at all when using a service.

- *Non-compliance* describes situations in which institutions do not comply to rules, laws and policies regarding privacy.

LINDDUN has been analyzed in empirical studies that tested how different threat models affect the traceability of different privacy threats. These studies showed that this threat model is easy to learn but still provides reliable results in comparison to experts. LINDDUN's threat trees have been considered useful in practice. LINDDUN itself provides a whole bottom-up privacy analysis pipeline as well, however, I only utilize their threat tree in my scenario [33].

2.6.3. Open-Loop Nature of Privacy

Privacy can only be guaranteed in closed systems which provide full control of their components. When components communicate with instances from different systems one has to trust those entities to follow the protocol participants have agreed on. Constraints that come along with these protocols can hardly be enforced but need to be trusted most of the time. They are called *open loop* constraints.

3. Related Work and Existing Privacy Analysis Approaches

Existing privacy analysis techniques have different advantages and disadvantages. This section discusses some approaches and highlights general problems with these techniques.

3.1. Privacy Risk Management

Privacy risk management is the general term which describes the handling of privacy risks within systems. From the management's perspective privacy risks consist of two factors: the first one is the likelihood of a problematic action, the second one the impact of this action. Multiplying those values results in the privacy risk that can be evaluated and eliminated [34].

Privacy Risk	**=**	**Likelihood of a problematic action**	**x**	**Impact of a problematic action**

The resulting privacy risk can be used by managers to evaluate different risks and decide which risks need to be eliminated. The elicitation of risks, their likelihood and impact are not part of this approach. This technique can be used on top of existing approaches.

3.2. Analyze Data Flow to Elicit Privacy Risks

Different data flow analysis techniques have been developed to track data flow and elicit privacy risks. The approach described by Lu and Li [35] includes different existing data flow analysis techniques such as "conditional flow identification" and "joint flow tracking". They implemented a system that analyzes Android application files for malicious data flow. This includes revealing contacts, call logs, browser history, SMS history, GPS or unique user IDs. A similar system for iOS applications has been developed by Egele and Kruegel [36]. Their system is able to detect data flow in compiled Objective-C binaries, similar to Lu's and Li's approach. Another interesting approach has been developed by Enck and Gilbert [37]. Their system can analyze data flow in Android applications in real-time, in contrast to the static approaches of Lu, Li, Egele and Kruegel. Enck and Gilbert's system *TaintDroid* can be run on productive devices in the background to spot malicious app requests or for testing purposes to see if an application is requesting specific information.

Analyzing data flow, as suggested by Lu and Li [35], Egele and Kruegel [36] and Enck and Gilbert [37], focuses on data sharers and data observers and data exchanged between them. However, these three approaches are optimized for mobile applications and only consider access to initial information sources, such as contact information but do not elicit privacy risks that can occur with data that has been exchanged with other systems or participants.

They do not consider what happens with this information outside of their scope. In many cases, it is necessary to exchange information for a service to be able to work as expected. Revealing private information is not always a privacy risk. Later on when data is exchanged with other partners or combined with other data sets privacy risks can occur as well which would not be covered by these approaches.

Another example for data flow analysis is the LINDDUN methodology, described by Wuyts, Scandariato and Joosen [32]. LINDDUN includes a privacy threat catalog but offers a data flow analysis technique as well. This approach uses a data flow diagram as a starting point to find privacy threats. A privacy threat catalog is then used to categorize each entity of the diagram into seven possible threat categories: *linkability*, *identifiability*, *non-repudiation*, *detectability*, *information disclosure*, *unawareness* and *non-compliance*. This methodology goes even further and describes a process to resolve privacy threats: The data flow analysis uses the threat catalogue to elicit privacy requirements and to be able to suggest solutions.

However, it is difficult to analyze complicated socio-technical systems using this approach, because it is focused on a bottom-up data flow analysis. Privacy risks often result from human interaction or interactions with different systems which makes it difficult for bottom-up analysis techniques to unveil these. Indeed, this drawback has been proven by Wuyts, Scandariato and Joosen in a set of extensive empirical studies [33].

3.3. Series of Open Questions to Elicit Privacy Risks

Hong, Ng, Lederer and Landay [29] developed an approach that uses a set of open questions, such as *Who are the users of the system?* and *What kinds of personal information are shared?* to stimulate engineers to think about consequences and possible privacy risks.

The big advantage of this approach is that open questions can help elicit unique privacy risks — but this approach involves risks as well: These questions can only be a rough guideline for finding privacy risks, they are not a straight-forward approach. They offer a framework for risk analysis but do not necessarily lead to a complete privacy risk analysis.

Approaches that involve specific requirements on their risk elicitation process and a specific pipeline would lead to more consistent and therefore more reliable results.

3.4. User-Level Privacy

3.4.1. Control and Feedback

Control and feedback [38] is an approach that gives users full control of their data. Users can decide whether or not which kind of data is revealed to entities, and they receive feedback whenever data is recorded or revealed. This means that privacy lies in the responsibility of each user. Control and feedback is a good mechanism to keep users up to date about sensor and data usage. The user is aware of ongoing processes and can decide if they accept certain capabilities. However, it can be annoying for users to decide this each time when using the system. Storing the preference of users might lead to unaware situations in which the user forgets that they allowed access to certain sensors. After all, this strategy helps to create awareness for privacy in direct user interaction. However, this cannot replace further and more in-depth analysis which handles privacy risks outside of the direct user scope, such as processing and sharing of data.

Figure 5 shows the four relevant categories: Capture, construction, accessibility and purpose.

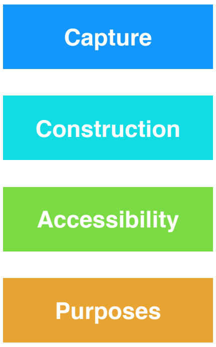
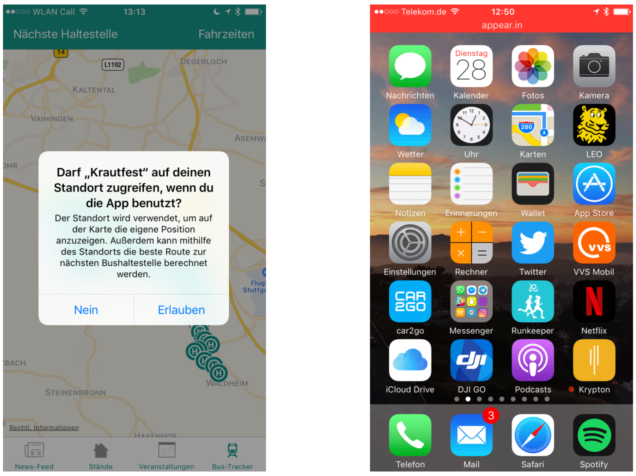


Figure 5: Control and Feedback: Users have full control what happens to their data.

Figure 6 shows a real-world example how *capture* and *purpose* are implemented in Apple's iOS operating system: The app on the left side needs to ask for permission to be able to access privacy-relevant information, such as GPS positions. The alert view gives an explanation which data is being accessed (*capture*), and how and why this data is being used (*purpose*). The screenshot on the right side indicates that data is captured in the background (in this case microphone) by showing the prominent red bar on the top of the screen. This kind of feedback ensures that the user does not forget and is aware of software collecting data in the background.



Darf „Krautfest" auf deinen Standort zugreifen, wenn du die App benutzt?

Der Standort wird verwendet, um auf der Karte die eigene Position anzuzeigen. Außerdem kann mithilfe des Standorts die beste Route zur nächsten Bushaltestelle berechnet werden.

Figure 6: Two real-world examples for *Control and Feedback*. On the left side considering *capture* and *purpose*. The alert view says: "Do you allow 'Krautfest' to access you location when using the app? The locating is being used to show your position on the map and to show the nearest bus stops". The screenshots on the right side indicates that audio is recoded in the background by showing the red bar in a prominent location.

3.5. General Problems of Existing Privacy Analysis Techniques

Existing privacy analysis techniques, like the ones mentioned above, do not consider the open-loop nature of many privacy-related issues: Many aspects of privacy are defined in laws, terms of use or privacy policies — and it is not possible to ensure that a user really read the policy and knows what can happen to their data.

Abstraction is another problem of current privacy analysis techniques: Many systems are embedded into complicated socio-technical environments. A technical bottom-up approach by looking at data flow does not always fulfill the purpose. Human interaction, internal components and interactions with other systems result in a complex system, so bottom-up analysis techniques are no longer objective. Additionally, it is very hard to verify if the company operates in conformity with their privacy policy. This is especially true for large companies with many departments. Therefore, the policy alone does not guarantee the confidentiality of the data, it must rather be enforced within the company as well as verified from the user's point of view.

3.6. Privacy in Health-Related Scenarios

Meingast, Roosta and Sastry [39] describe emerging privacy issues with health-related information. However, they do not limit their work to be applied to heath-related privacy issues. They consider different fields that benefit from these results as well, including financial services or internet shops. They present existing solutions and future work to solve these issues, such as "*clear attributes for role based access*", "*encryption*", "*authentication mechanisms*", "*policy development*", "*rules on patients privacy at home*" and "*data mining rules and technological measures*” [39].

Kaletsch and Sunyaev [40] analyzed different cloud-based health scenarios and found multiple common privacy threats caused by different systems. They found out that social features, selling medical information, advertising and analytics are the most common privacy threats among cloud-based health services. They also started to develop a privacy framework for health platforms to be able to ensure the patients' privacy.

4. STPA Extension for Privacy Risk Analysis: STPA-Priv

STPA is an analysis technique that has originally been developed to support the analysis of safety risks within systems. However, STPA is not limited to safety-related topics: A new sub-type of STPA, called STPA-Sec, has been optimized to help finding security risks within systems [20].

Shapiro extended STPA-Sec to be used for the elicitation of privacy risks [6, 19]. His proposed extension is called STPA-Priv. It combines the existing advantages of STPA with an extension for privacy analysis. This includes the top-down principle of STPA to be able to handle complex socio-technical systems. Parts of the process have been renamed, but the overall steps of STPA stay the same.

This chapter explains how the extension of STPA for privacy looks like. Focusing on privacy solely is not straight-forward due to the huge overlapping of privacy and security. To ensure privacy in a system it is necessary to ensure security in the first place — technical security aspects like encryption are the foundation for preventing all privacy risks [29].

4.1. Definitions and Renaming

4.1.1. Losses and Accidents

Loss or *accidents* in traditional STPA are always related to a loss of human life, injuries or destruction of expensive hardware. However, privacy violations often do not lead to accidents which threaten human life but could lead to embarrassing, awkward and adverse situations for individuals, or emotional damage in general. Even more threatening for individuals are the consequences of identity theft. This is why losses and accidents are (re)named to *adverse consequences* in respect to privacy [34].

4.1.2. Unsafe Control Actions

Unsafe control actions in STPA are commands sent between participants of a system, that can lead to a hazardous system state. Adverse consequences are also considered a result of unsafe control actions. Accordingly *unsafe control actions* are (re)named to *privacy-compromising control actions* [6].

4.1.3. Safety Constraints

Safety constraints in STPA-Priv are simply called privacy constraints. Their functionality remains the same.

4.1.4. Privacy Risk Model

Discussing which risk model should be the foundation of the privacy risk analysis is essential. I presented two risk models in section 2.6: A general approach called *Calo's subjective/objective privacy harms* and the more detailed *LINDDUN privacy threat catalog*. Privacy risk models draw a framework under which a common understanding of the wide field of privacy is possible.

4.1.5. Open-Loop Constraints

Open-loop constraints in general are constraints that cannot be ensured or enforced in a system. A common example for open-loop constraints are privacy policies which a user needs to read before using a service — of course it cannot be enforced that a user has read it, but it might contain important information regarding the handling of sensitive data. The same problem occurs the other way around: how can a user know that a provider of a service is using the data only in a manner that is described in the privacy policy? Both scenarios can lead to adverse consequences for the user, however, there is not much one can do about it. Another example are privacy-related laws which enforce companies to handle data in a specific manner, however, it is not always possible to check how data is really handled by companies or governments. What happens with data in general that is provided to an entity? How can one ensure that the data is only being used in the promised manner? The only solution in this case is: trust no one, keep your data for yourself.

After all, my system could contain an unlimited number of edge cases and participants with more or less controllable constraints [29]. Therefore I cannot ensure perfect privacy but aim for a system with reasonable privacy.

4.1.6. Participants and their Relationships

An important part of privacy risk analysis are the different participants and their relationships: Do participants trust each other? Who are data providers and who are data observers? Is it a one-sided relationship with an unbalanced level of trust? It can help to arrange all participants in a two-dimensional grid, representing strong or weak commercial or social interests of participants, as illustrated in Figure 7. These assessments can later help in the process of STPA-Priv to elicit privacy risks and create the control structure diagram.

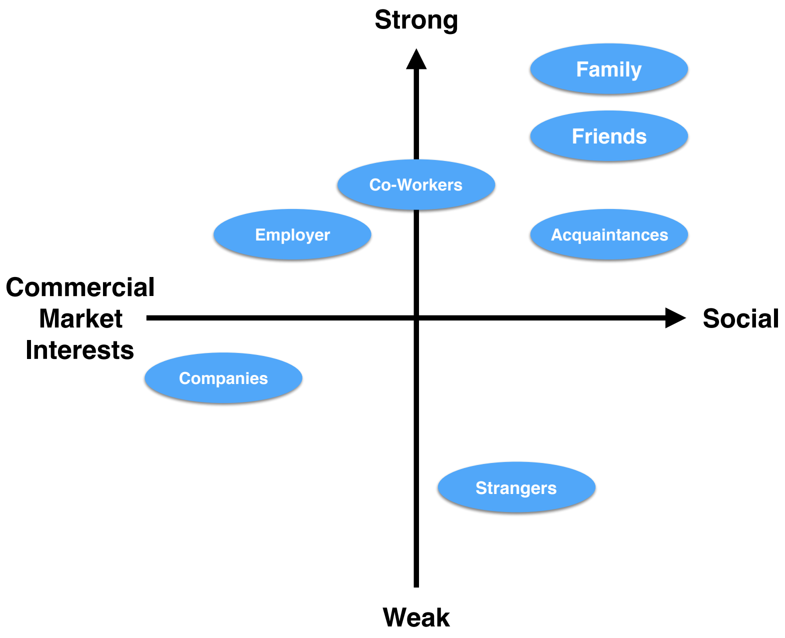


Figure 7: Different kinds of relationships on a two-dimensional grid: Are relationships more of a social or a commercial interest? Are relationships weak or strong? Classifying relationships can help to identify possible risks later on.

4.1.7. Data Flow

Thinking about exchanged data types helps to identify adverse consequences later on. Ask yourself questions like *What kind of data is exchanged between participants?* or *Is data transferred completely, or are data sets filtered beforehand?* or *What information could be reconstructed from data sets?* or *To whom could the information be forwarded?*. These questions are similar to the approach showed in section 3.3 where open questions are being used to help engineers to determine privacy risks. In my case I am using these questions as a starting point for eliciting adverse consequences. Figure 8 shows different kinds of data sets that could be transferred between participants.

Full Data Set
Partial Data Set
Specific Data Set

Figure 8: Data flow: What kind of data is transferred to other participants? Classifying exchanged data can help identify adverse consequences.

4.1.8. Control Structure Diagrams

Control structure diagrams in privacy-sensitive systems can be used the same way as in traditional STPA. However, to prevent misunderstandings changes to the terminology would be useful. Sensors that provide feedback to the controller in traditional STPA are not always *hardware sensors* in STPA-Priv. Especially in eHealth scenarios with GPS sensors and heart rate sensors this can lead to confusion, because those sensors are not sensors in the context of control loops. It would be useful to rename control loop sensors in *feedback mechanisms*. This has two positive effects: First, the risk of misunderstanding the meaning of “sensor” is eliminated, and second, it is more clear that a feedback mechanism in STPA-Priv is not limited to hardware components.

The second change refers to the terminology of *actuators*. This is a very hardware-specific concept as well. Actuators are devices that convert electric signals into mechanical movement. A renaming to *initiator* would make sense for privacy. Additionally there is sometimes no additional *initiator* between a controller and the process itself when it comes to privacy-aware systems, because the controller is the initiator.

4.2. STPA-Priv Extension for XSTAMPP

The software tool XSTAMPP offers support for STPA-Priv via a plugin. Before applying STPA-Priv to a scenario this extension should be installed in order to support the process. The plugin can be downloaded from XSTAMPP’s sourceforge webpage[2] [41]. The required file is called ***updatesite.zip***. Unzip the file to extract its content.

After downloading the plugin, open XSTAMPP and go to ***Help*** → ***Install New Software***. From there select *Add* and click on *Local*. Not you can select the previously downloaded folder ***updatesite***. Select STPA-Priv and XSTPA-Priv and click on ***Next***. Accept the necessary license agreements and follow the procedure. In the end, after XSTAMPP has been restarted you should see the new extension when clicking on ***File*** → ***Create New Project***.

4.3. Methodology of STPA-Priv

STPA originally has been designed to analyze systems and their environment including unpredictable factors such as humans. The goal of STPA-Priv is now to utilize this methodology for privacy risk analysis. The general process of STPA-Priv includes similar steps as in STPA:

1) Fundamental analysis in which privacy analysts identify adverse consequences, hazardous system states, privacy constraints and draw the high-level privacy control structure diagram.

2) STPA-Priv step 1: Identify privacy-compromising control actions in the control structure diagram in four different categories: not providing causes hazard, providing causes hazard, wrong timing or order causes hazard or applying too long/stopping too soon causes hazard.

3) STPA-Priv step 2: Causal analysis in which the privacy analysts identify causal factors and scenarios for each privacy-compromising control action to understand how they could occur. These steps and their results are depicted in Figure 9.

[2] https://sourceforge.net/projects/stpa-for-privacy-analysis/files/?source=navbar

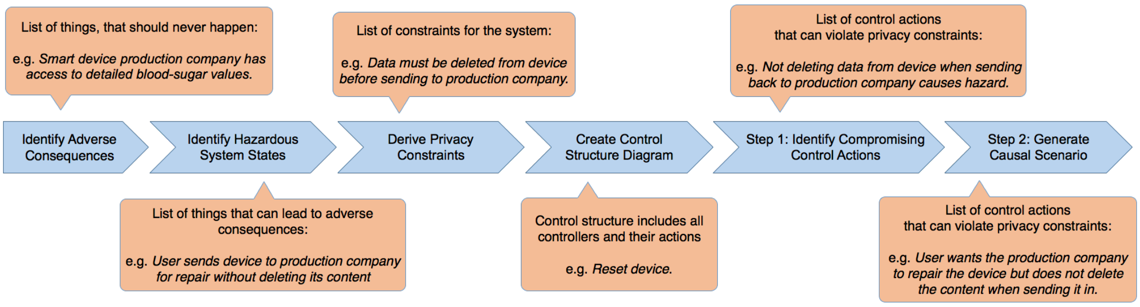

List of things, that should never happen:
e.g. *Smart device production company has access to detailed blood-sugar values.*
List of constraints for the system:
e.g. *Data must be deleted from device before sending to production company.*
List of control actions that can violate privacy constraints:
e.g. *Not deleting data from device when sending back to production company causes hazard.*
Identify Adverse Consequences
Identify Hazardous System States
Derive Privacy Constraints
Create Control Structure Diagram
Step 1: Identify Compromising Control Actions
Step 2: Generate Causal Scenario
List of things that can lead to adverse consequences:
e.g. *User sends device to production company for repair without deleting its content*
Control structure includes all controllers and their actions
e.g. *Reset device.*
List of control actions that can violate privacy constraints:
e.g. *User wants the production company to repair the device but does not delete the content when sending it in.*

Figure 9: The STPA-Priv pipeline and examples for results of each step. The pipeline starts with identifying adverse consequences, continues with hazardous system states, deriving privacy constraints, creating the control structure diagram, identifying privacy compromising control actions and finally generating causal scenarios.

4.3.1. Identify involved participants and their relationships

Before starting to analyze a system one has to be aware of all participants in the scenario. This helps identify all relationships and find adverse consequences later. The first set of participants can be taken straight from the scenario. If participants have been disregarded initially, they can still be added during the process iteratively. It makes sense to arrange participants in a two-dimensional grid, as shown in Figure 7.

4.3.2. Identify Adverse Consequences

Finding and defining adverse consequences in the system is the first important step of STPA-Priv. Finding adverse consequences requires experts that are aware of the features provided by the system. However, it is not necessary to know the implementation of each component, since STPA is a top-down approach. Adverse consequences are those consequences that should not occur in my system. Stakeholders need to agree on adverse consequences that should be prevented [8]. A privacy framework, such as *LINDDUN privacy threat tree catalog* [32] or *Calo's subjective/objective privacy harms* [6, 31], is used as a starting point — threats described in these frameworks need to be applied to the scenario to generate adverse consequences.

Elaborating adverse consequences is the counterpart to accidents in standard STPA. It is important to identify involved participants and their relationships and to select a privacy framework first to be aware of the scenario and have a common understanding of privacy.

Privacy risk models, such as the LINDDUN privacy threat model, help analysts elicit adverse privacy consequences. The standard LINDDUN methodology derives privacy threat scenarios from data flow diagrams including *entities*, *data stores*, *data flows* and *processes* [32]. With STPA-Priv I want to prevent this bottom-up approach which does not fit for complicated socio-technical environments. This is why I concentrate on participants, their relationships and knowledge about my scenario to find adverse consequences: I analyze how *linkability*, *identifiability*, *non-repudiation*, *detectability*, *information disclosure*, *unawareness* and *non-compliance* can be present in my scenario, for participants and relationships.

4.3.3. Identify Hazardous System States

Each adverse consequence can be triggered by one or more system states together with environmental conditions of the system. They are called hazardous system states. Hazards are system states that are under the system's control, whereas adverse consequences themselves are not controllable. This is why I want to prevent hazardous system states, because their appearance is under the system's control — and preventing hazardous system states prevents adverse consequences.

4.3.4. Derive Privacy Constraints from Identified System Hazards

Privacy constraints are constraints that ensure that hazardous system states are not occurring. Each hazardous system state must be covered by a privacy constraint. Privacy constraints are a positive formulation of hazards in order to prevent them.

4.3.5. Control Structure Diagram

Control structure diagrams in standard STPA highlight controllers, actuators, processes, sensors, control actions, and feedback within systems. When it comes to privacy it is not always possible to provide feedback on processes. I am speaking of open-loop controllers in such cases.

Overall there are three different levels of control loops that can be applied to privacy scenarios:

1. *High-level control loops* represent the general feedback loop which ensures that systems are conform to privacy laws and policies. This applies to many systems: The whole system is controlled by laws, defining which strategies and policies are legal in respect to privacy. The actuator that actually starts the process can differ from system to system. In many cases people agree to use a service in order to improve or analyze their financial situation (e.g., FinTec), their health condition (e.g., eHealth) or the overall improvement of workflows and living conditions (e.g., Internet of Things). The *process* itself is represented while using the system. And the sensor is represented by the awareness for privacy in this system: Each user needs to be aware of privacy and needs to check if the device usage violates privacy laws and how the privacy policy is shaped. The feedback which comes back to the system can be as simple as not buying or not using the product anymore if any violations have been detected. Figure 10a shows how high-level control loops can look like.

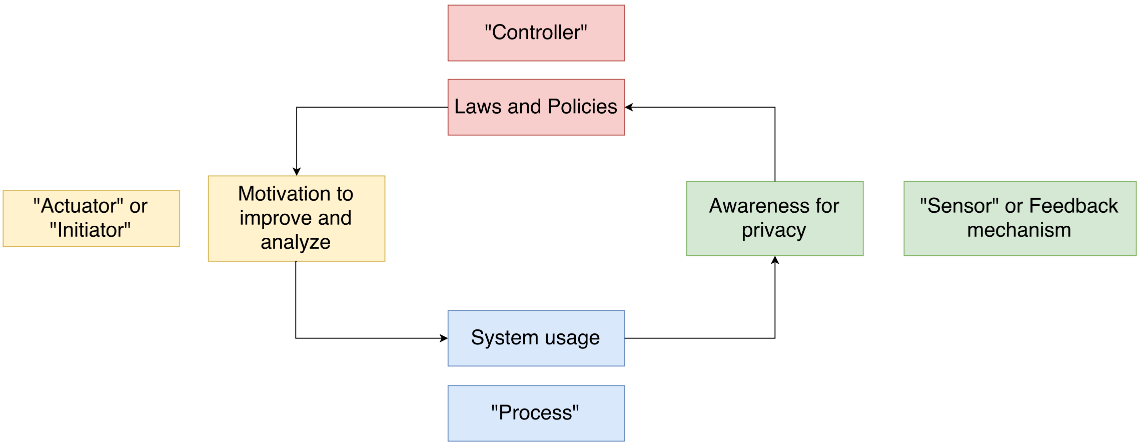

"Controller"
Laws and Policies
"Actuator" or "Initiator"
Motivation to improve and analyze
Awareness for privacy
"Sensor" or Feedback mechanism
System usage
"Process"

Figure 10a: A general example for high-level control loops that is part of many privacy-related systems.

2. *User-level control loops* represent control loops that directly interact with the user. These control loops implement technologies similar to *control and feedback* from section 3.4. This ensures that the user is aware of sensor data that is currently being accessed, data that is transmitted or that the user has read the privacy policy that comes with this system. Figure 10b illustrates how user-level control loops can look like.

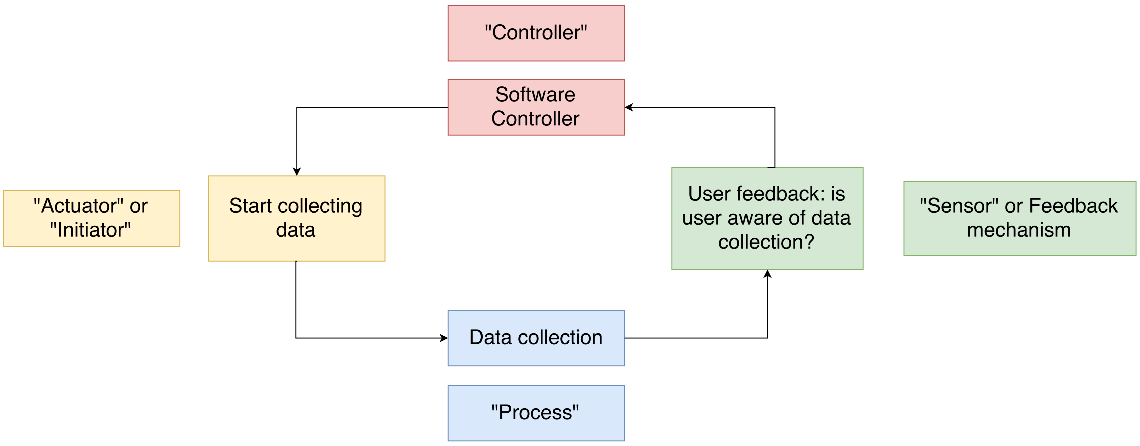

"Controller"
Software Controller
"Actuator" or "Initiator"
Start collecting data
User feedback: is user aware of data collection?
"Sensor" or Feedback mechanism
Data collection
"Process"

Figure 10b: A general example for user level control loops.

3. *Technical control loops* are control loops that ensure the technical requirements of the system. This includes that certain (raw) data sets do not leave the system or only if filtered beforehand. The example shows how the control loop is structured. It is the responsibility of specific data controllers to provide data sets for external stakeholders. Before these data sets are provided they are processed through the feedback loop. Sensitive parts of these data sets are removed. All data sets are checked by a feedback mechanism for their sensitivity. If everything is fine the data can be transferred. The diagram in Figure 10c shows how technical control loops can look like.

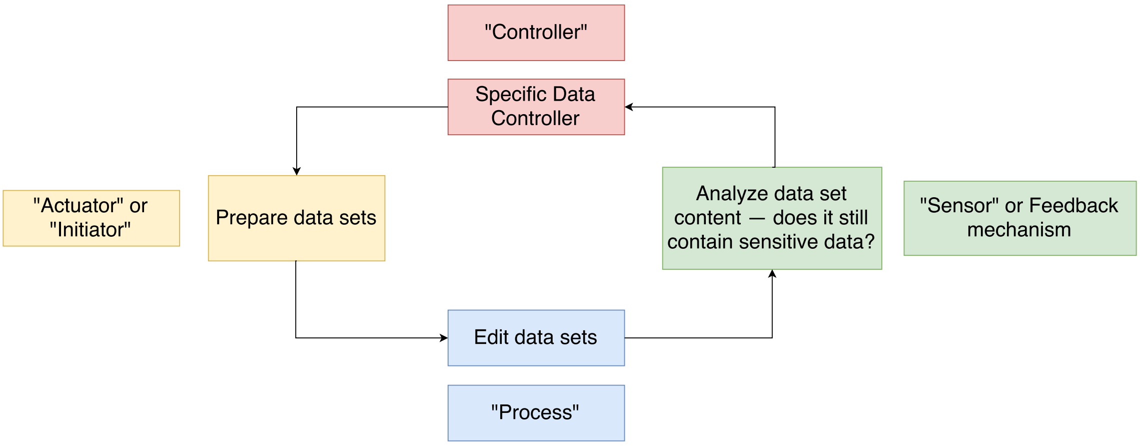

"Controller"
Specific Data Controller
"Actuator" or "Initiator"
Prepare data sets
Analyze data set content — does it still contain sensitive data?
"Sensor" or Feedback mechanism
Edit data sets
"Process"

Figure 10c: A general example for technical control loops. Data sets are prepared to be shared by removing sensitive content.

4.3.6. Step 1: Identify Privacy Compromising Control Actions (UCAs)

Privacy-compromising control actions are control actions that can violate privacy constraints when being executed. The goal of this step is to find control actions that would violate privacy constraints and therefore can lead to hazardous system states. In the end, these privacy-compromising control actions are the flaw of the system and need to be prevented by engineers. Privacy-compromising control actions show where the system could result in a hazardous system state. Later on, in *step 2*, causal scenarios are generated for each control action.

Each privacy constraint is enforced by a controlling component. All controlling components are part of the control structure. A control action could lead to a corresponding hazardous system state. Malfunctions of privacy-compromising control actions can be classified in one of those four categories:

- *Not providing the control action*, when it should be provided causes a hazard.
- *Providing the control action*, when it should not be provided causes a hazard.
- *Providing the control action too early*, *too late or in wrong order* causes a hazard.
- *Stopping the control action or applying it too long* causes a hazard.

When looking at the privacy constraints one has to find the appropriate control action from the control structure that is responsible for ensuring the privacy constraint.

4.3.7. Step 2: Generating Causal Scenarios

The previous step generated a list of privacy-compromising control actions that can violate privacy constraints and therefore potentially cause hazardous system states. They describe *what* could go wrong. The last step of STPA-Priv concludes scenarios that describe *how* a privacy-compromising control action might be executed. The goal is to find causal scenarios in the system that can lead to privacy-compromising system states. This is not limited to simple components, but can occur in conjunction with components and control actions within the whole socio-technical system. This is also often referred to as *worst case scenario*. One can look at hazards that are caused by these control actions to find causal scenarios.

Control actions that can be referred to a causal scenario require a risk management response. This includes adding systematic test cases that can reproduce the causal scenario.

5. Applying STPA-Priv to Internet of Things Scenario

The goal of this thesis is to apply STPA-Priv to an Internet of Things scenario and analyze its applicability this way. This chapter describes which internet of things scenario has been selected and why — I then apply STPA-Priv to this scenario. I use the tool *XSTAMPP* with its STPA-Priv extension as support while executing STPA-Priv. Click on ***File*** → ***Create New Project*** → ***STPA-Priv*** in order to get started.

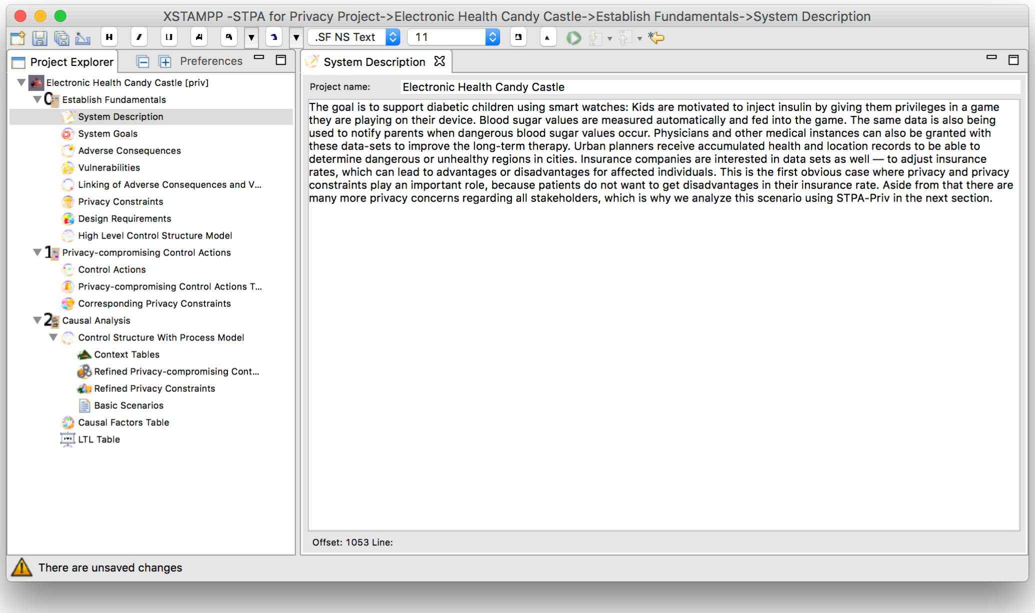
5.1. Internet of Things Scenario

Many scenarios would fit well for an evaluation of STPA-Priv – smart TVs, instant messengers, smart home or eHealth scenarios. Almost every device or service collects, processes and transmits data and would therefore be a good scenario for STPA-Priv. I decided to apply STPA-Priv to an eHealth scenario, because it covers a wide range of topics and difficulties regarding privacy: extremely sensitive data of different types and many stakeholders with different interests.

Knöll developed an eHealth system together with the Olgahospital, a children's hospital in Stuttgart, Germany [25]. I am using the enhanced version of this scenario which has been developed by Stach [42]. The general idea is to use the serious game *Candy Castle* to motivate children to follow the therapy suggested by their physician. Patient’s parents, their physician, insurance companies and urban planners are part of this system as well. All of them have different expectations and requirements for the product and data sets. This makes it interesting to analyze. The overall goal is to support diabetic children using smart devices: Children are motivated to inject insulin by giving them privileges in the serious game which they are playing on their device. Blood sugar values are measured automatically and fed into the game. All blood sugar data points are then stored along with the current location and a timestamp. The same data is also being used to notify parents when dangerous blood sugar values occur. Physicians and other medical instances can also be granted with these data sets to improve the long-term therapy. Urban planners receive accumulated health and location records to be able to determine dangerous or unhealthy regions in cities.

Insurance companies are interested in data sets as well — to adjust insurance rates, which can lead to advantages or disadvantages for affected individuals. This is the first obvious case where privacy and privacy constraints play an important role, because patients do not want to get disadvantages in their insurance rate. Aside from the fact that there are many more privacy concerns regarding all stakeholders. Not every stakeholder requires access to all data sets, the access must be restricted. Most of the captured data points are highly sensitive regarding privacy. This is why I analyze the given scenario using STPA-Priv in the next section.

The software XSTAMPP offers to store general information on the system in its section *System Description*, as shown in Figure 11. More specific goals of the system are put in the section *System Goals*, as shown in Figure 12.


XSTAMPP -STPA for Privacy Project->Electronic Health Candy Castle->Establish Fundamentals->System Description
.SF NS Text
11
Project Explorer
Preferences
Electronic Health Candy Castle [priv]
0 Establish Fundamentals
System Description
System Goals
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and V...
Privacy Constraints
Design Requirements
High Level Control Structure Model
1 Privacy-compromising Control Actions
Control Actions
Privacy-compromising Control Actions T...
Corresponding Privacy Constraints
2 Causal Analysis
Control Structure With Process Model
Context Tables
Refined Privacy-compromising Cont...
Refined Privacy Constraints
Basic Scenarios
Causal Factors Table
LTL Table
System Description

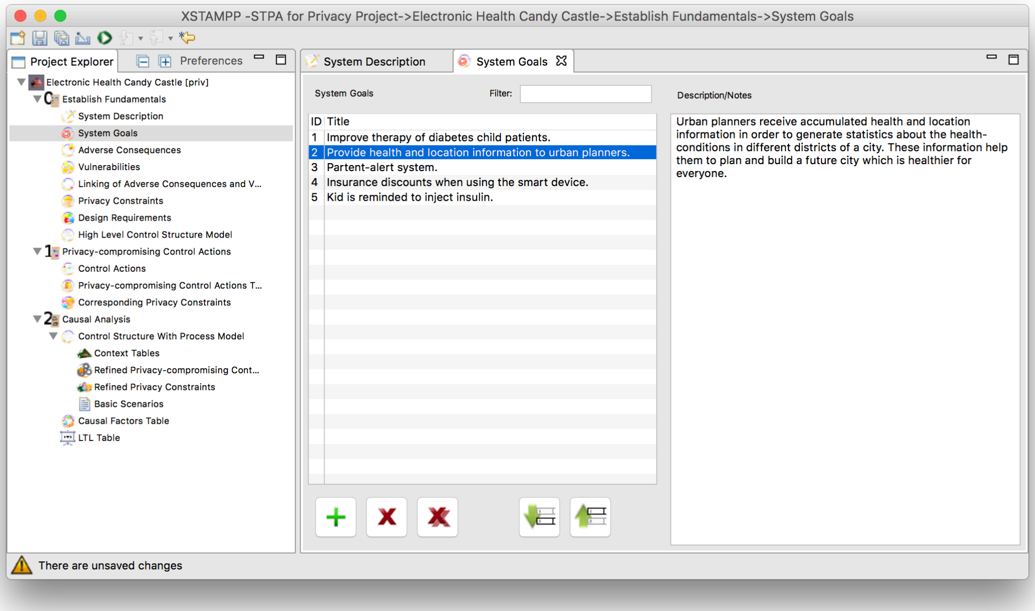
Project name: Electronic Health Candy Castle

The goal is to support diabetic children using smart watches: Kids are motivated to inject insulin by giving them privileges in a game they are playing on their device. Blood sugar values are measured automatically and fed into the game. The same data is also being used to notify parents when dangerous blood sugar values occur. Physicians and other medical instances can also be granted with these data-sets to improve the long-term therapy. Urban planners receive accumulated health and location records to be able to determine dangerous or unhealthy regions in cities. Insurance companies are interested in data sets as well — to adjust insurance rates, which can lead to advantages or disadvantages for affected individuals. This is the first obvious case where privacy and privacy constraints play an important role, because patients do not want to get disadvantages in their insurance rate. Aside from that there are many more privacy concerns regarding all stakeholders, which is why we analyze this scenario using STPA-Priv in the next section.

Offset: 1053 Line:

There are unsaved changes

Figure 11: XSTAMPP contains a section to describe the system. The whole pipeline of STPA-Priv is visible on the left side of the software. The terminology has already been updated to be consistent with the terminology in this document.


XSTAMPP -STPA for Privacy Project->Electronic Health Candy Castle->Establish Fundamentals->System Goals
Project Explorer
Preferences
Electronic Health Candy Castle [priv]
0 Establish Fundamentals
System Description
System Goals
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and V...
Privacy Constraints
Design Requirements
High Level Control Structure Model
1 Privacy-compromising Control Actions
Control Actions
Privacy-compromising Control Actions T...
Corresponding Privacy Constraints
2 Causal Analysis
Control Structure With Process Model
Context Tables
Refined Privacy-compromising Cont...
Refined Privacy Constraints
Basic Scenarios
Causal Factors Table
LTL Table
System Description
System Goals
System Goals
Filter:
Description/Notes
There are unsaved changes

ID	Title
1	Improve therapy of diabetes child patients.
2	Provide health and location information to urban planners.
3	Partent-alert system.
4	Insurance discounts when using the smart device.
5	Kid is reminded to inject insulin.

Urban planners receive accumulated health and location information in order to generate statistics about the health-conditions in different districts of a city. These information help them to plan and build a future city which is healthier for everyone.

Figure 12: System goals can be entered in this section of XSTAMPP. System goals are general descriptions of purposes of the system. A detailed description or additional notes can be added on the right side.

5.2. Applying STPA-Priv to eHealth-Scenario to Elicit Privacy Risks using XSTAMPP

5.2.1. Identify Involved Participants and their Relationships

It is important to identify participants of my scenario: This helps us define the system and be clear about consequences and relationships between individuals. Involved participants can be either data sharers or data observers. The relationship can be either out of social or commercial interest with different interests to protect disclosed information, which is reflected in the trust the data sharer has towards other participants.

I now want to find participants in my system which are directly related to the main character (diabetes child) or data he or she is sharing. Identifying participants is an iterative approach. New participants can be added during the process of STPA-Priv. [29]

From the perspective of the main character I identified five main participants in the system: The parents, the physician, the company that produced the smart device, urban planners and the health insurance company. This can be extended even further when considering third parties that might be impacted by my system and could play an important role . Their interests and level of trust are interesting for us to rate their relevance in the system, their attitude towards privacy and potential risks [29]. Later, this helps us to find related adverse consequences. The visualization of participant in the two-dimensional grid can be seen in Figure 13.

I used different strategies to identify participants. The first and obvious source is the scenario itself which already provides a list of participants. It is also useful to take a look at potential malicious data observers and impacts of the system beyond its borders. Participants that are involved in developing and maintaining the hardware and software that is being used to collect, share and store private data are interesting to look at as well [29].

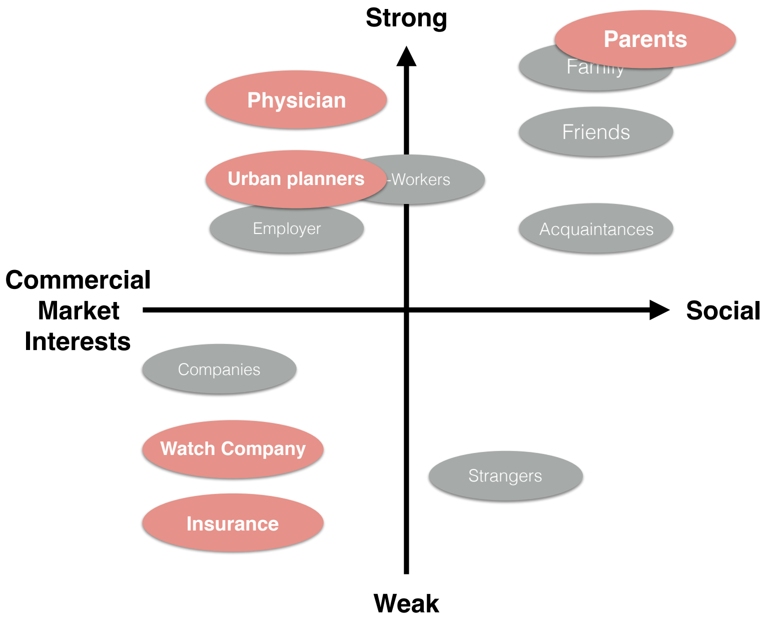


Figure 13: This figure shows the different participants of the scenario (red) in a two-dimensional grid to visualize type and strength of relationship from the perspective of the data provider. The grey entities are displayed for reference.

5.2.2. Selecting the Privacy Risk Model

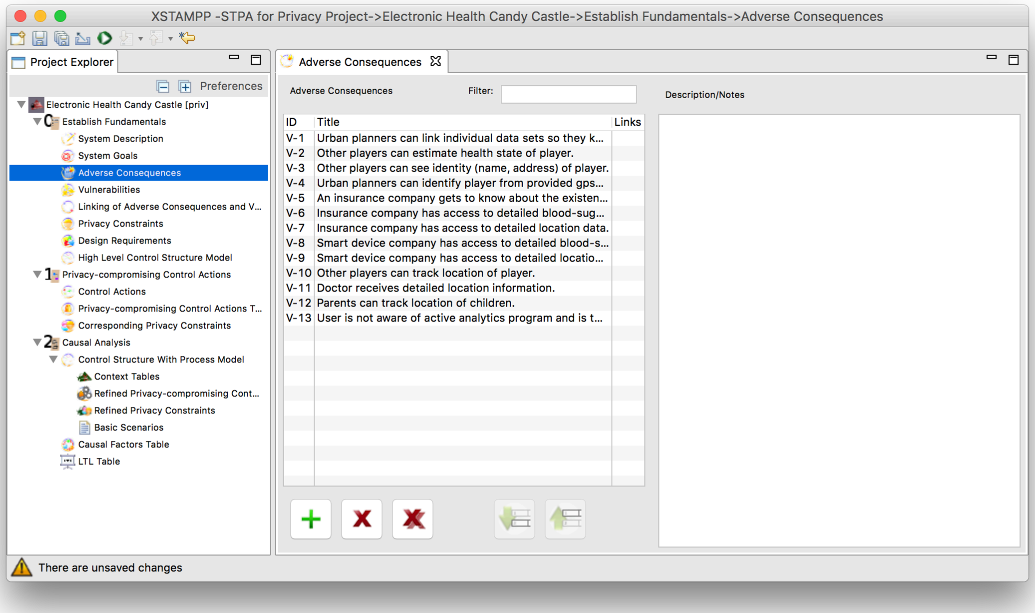
When discussing privacy it is important to have the same understanding of privacy. This is especially important when designing a privacy-aware system where different people are involved. In my case I selected the LINDDUN privacy threat tree, which is part of the LINDDUN privacy threat modeling methodology. The threat tree offers a wide range of privacy issues, arranged in seven categories with many sub-categories: *linkability*, *identifiability*, *non-repudiation*, *detectability*, *information disclosure*, *unawareness* and *non-compliance*. The threat tree helps us to identify adverse consequences in the next step.

5.2.3. Identify Adverse Consequences

Applying the first step of STPA-Priv to the scenario seemed unfamiliar and difficult in the beginning. Focussing on privacy issues instead of security or safety hazards required restructuring the mindset to use this approach. The first thing to ensure is to choose the right level of abstraction: Adverse consequences are not only about findings hazards related to the unintentional disclose of private data. It is required to go one step further and find concrete instances of adverse consequences which could be triggered by an unintentional disclosure of information.

The main problem I faced is that the field of privacy and possible adverse consequences resulting from privacy vulnerability is huge, whereas each concrete system with its concrete scenario is always a very special use case. This requires special knowledge for this specific problem and a general approach might not cover all possible consequences. The LINDDUN threat tree was a great help to be able to consider different privacy threats: I was able to identify two adverse consequences on the field of *linkability,* which is the first category of the LINDDUN privacy threat catalog. The first adverse privacy consequence is that urban planners can link individual data sets so they know that they come from the same patient. The second one regarding *linkability* is when other players can estimate the health state of a player, which belongs to the LINDDUN category of *information disclosure* as well. The next category covers issues with identifiability: I found two adverse consequences in this field. The first one is that other players can see the identity (name, address) of a player, which is part of the unawareness category as well, and the second one making it possible for urban planners to identify players from provided location information and health data sets. I did not find any adverse consequences in the field of non-repudiation, but in the next category which is about *detectability*. Insurance companies could get to know about the diabetes disease of a player without being involved in the process. The meta information that the person is using the app is already enough that the insurance companies know that the child suffers from diabetes. This could lead to a rejection when registering for a new insurance. I was able to identify most adverse consequences in the next category which covers *information disclosure*: Insurance company has access to detailed blood sugar values or detailed location data; the smart device company has access to detailed blood sugar values or location data; the physician receives detailed location information; other players can track the location of the player or the parents can track the location of their child. The last category of the LINDDUN threat tree where I found adverse consequences is *unawareness*: This occurs when the user is not aware of active analytics program and is therefore suspect to surveillance.

All adverse consequences with their appropriate LINDDUN category are documented in Figure 14 using XSTAMPP and Table 4 for better readability.



XSTAMPP -STPA for Privacy Project->Electronic Health Candy Castle->Establish Fundamentals->Adverse Consequences

ID	Title	Links
V-1	Urban planners can link individual data sets so they k...	
V-2	Other players can estimate health state of player.	
V-3	Other players can see identity (name, address) of player.	
V-4	Urban planners can identify player from provided gps...	
V-5	An insurance company gets to know about the existen...	
V-6	Insurance company has access to detailed blood-sug...	
V-7	Insurance company has access to detailed location data.	
V-8	Smart device company has access to detailed blood-s...	
V-9	Smart device company has access to detailed locatio...	
V-10	Other players can track location of player.	
V-11	Doctor receives detailed location information.	
V-12	Parents can track location of children.	
V-13	User is not aware of active analytics program and is t...	

There are unsaved changes

Project Explorer
Preferences
Electronic Health Candy Castle [priv]
0 Establish Fundamentals
System Description
System Goals
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and V...
Privacy Constraints
Design Requirements
High Level Control Structure Model
1 Privacy-compromising Control Actions
Control Actions
Privacy-compromising Control Actions T...
Corresponding Privacy Constraints
2 Causal Analysis
Control Structure With Process Model
Context Tables
Refined Privacy-compromising Cont...
Refined Privacy Constraints
Basic Scenarios
Causal Factors Table
LTL Table
Adverse Consequences
Adverse Consequences
Filter:
Description/Notes

Figure 14: Adverse consequences are added in the appropriate section of XSTAMPP.

Adverse Privacy Consequences	LINDDUN Category	Hazardous System States
Urban planners can link individual data sets so they know that they come from the same player.	**Linkability**	Data sets submitted to urban planners include information about player.
		Data sets submitted to urban planners include pattern that can identify individuals.
Other players can estimate health state of player.	**Linkability, Information Disclosure**	High score allows assumptions on health state.
Other players can see identity (name, address) of player.	**Identifiability, Unawareness**	High scores include personal information of player.
Urban planners can identify player from provided GPS and health data.	**Identifiability**	Data sets submitted to urban planners include information about player.
		Data sets submitted to urban planners include pattern that can identify individuals.
An insurance company gets to know about the existence of the diabetes disease of the player.	**Detectability**	The system reveals who is using the app.
Insurance company has access to detailed blood sugar values.	**Information Disclosure**	Detailed blood sugar values are sent to insurance company as part of the general therapy data.
		User decides to stop using the device and sends it back to the insurance company without deleting its content.
Insurance company has access to detailed location data.	**Information Disclosure**	Detailed location data is sent to insurance company as part of the general therapy data.
		User decides to stop using the device and sends it back to the insurance company without deleting its content.
		High score allows assumptions on health state.
Smart device company has access to detailed blood sugar values.	**Information Disclosure**	Analytics data includes detailed blood sugar values.
		User sends device to company for repair without deleting its content.
Smart device company has access to detailed location data.	**Information Disclosure**	Analytics data includes detailed location data.
		User sends device to company for repair without deleting its content.

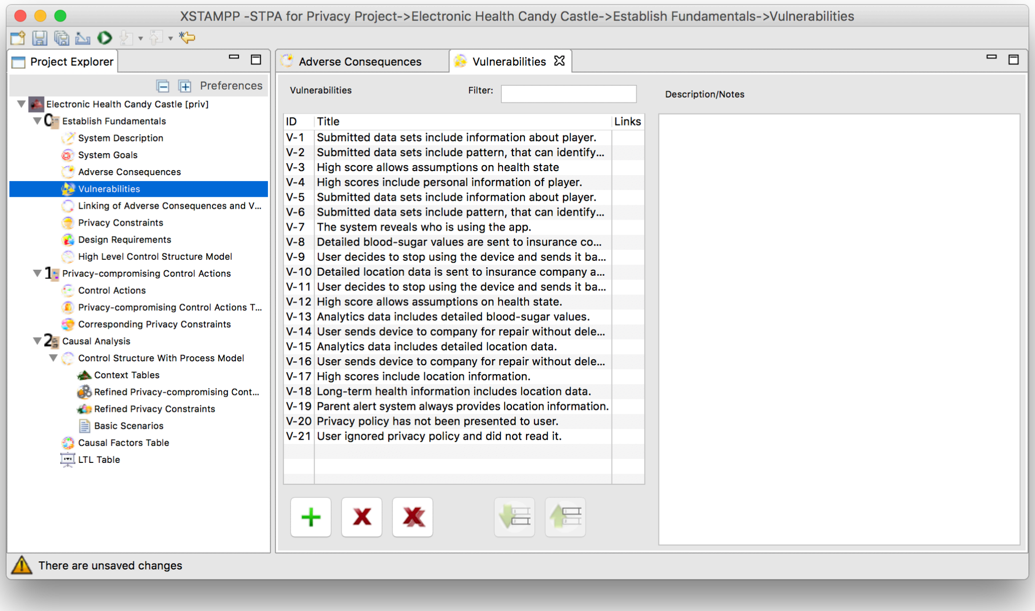
Adverse Privacy Consequences	LINDDUN Category	Hazardous System States
Other players can track location of player.	**Information Disclosure**	High scores include location information.
Physician receives detailed location information.	**Information Disclosure**	Long-term health information includes location data.
Parents can track location of children.	**Information Disclosure**	Parent alert system always provides location information.
User is not aware of active analytics program and is therefore suspect to surveillance.	**Unawareness**	Privacy policy has not been presented to user.
		User ignored privacy policy and did not read it.

Table 4: Adverse consequences and their hazards in the scenario. Each adverse consequence is categorized according to the LINDDUN privacy threat tree. This helps identify adverse consequences.

5.2.4. Identify Hazardous System States

The goal of this step is to identify hazardous system states that are related to the adverse consequences from the previous step. The idea is that adverse consequences themselves cannot be prevented, but the *triggers* that lead to these consequences can be prevented. These *triggers* are called hazardous system states, because systems in these states can potentially lead to adverse consequences: The first adverse consequence "*Urban planners can link individual data sets so they know that they come from the same player"* has two different hazards. The first one occurs when data sets that are submitted to the urban planners contain information about the player and the second one if the data sets include any kind of pattern that could identify individuals. The next adverse consequence "*Other players can estimate health state of player"* can be reached from the hazard "*High score allows assumptions on health state*".

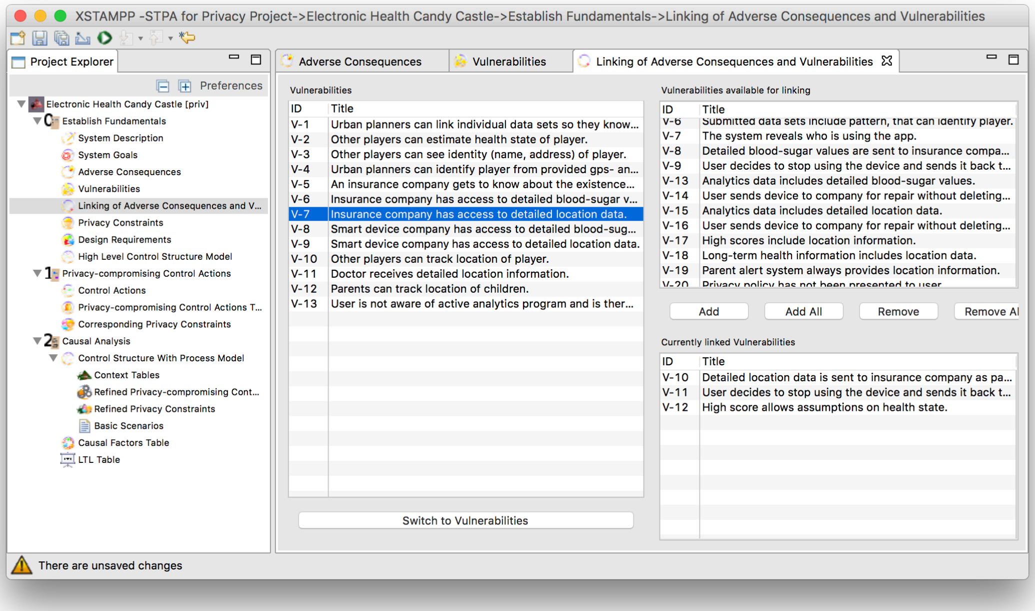
All hazardous system states have been added in the appropriate section of XSTAMPP, as showed in Figure 15. Hazards and adverse consequences are then linked with each other, as shown in Figure 16 and Table 4.


XSTAMPP -STPA for Privacy Project->Electronic Health Candy Castle->Establish Fundamentals->Vulnerabilities
Project Explorer
Preferences
Electronic Health Candy Castle [priv]
0 Establish Fundamentals
System Description
System Goals
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and V...
Privacy Constraints
Design Requirements
High Level Control Structure Model
1 Privacy-compromising Control Actions
Control Actions
Privacy-compromising Control Actions T...
Corresponding Privacy Constraints
2 Causal Analysis
Control Structure With Process Model
Context Tables
Refined Privacy-compromising Cont...
Refined Privacy Constraints
Basic Scenarios
Causal Factors Table
LTL Table
Adverse Consequences
Vulnerabilities
Filter:
Description/Notes
There are unsaved changes

Vulnerabilities

ID	Title	Links
V-1	Submitted data sets include information about player.	
V-2	Submitted data sets include pattern, that can identify...	
V-3	High score allows assumptions on health state	
V-4	High scores include personal information of player.	
V-5	Submitted data sets include information about player.	
V-6	Submitted data sets include pattern, that can identify...	
V-7	The system reveals who is using the app.	
V-8	Detailed blood-sugar values are sent to insurance co...	
V-9	User decides to stop using the device and sends it ba...	
V-10	Detailed location data is sent to insurance company a...	
V-11	User decides to stop using the device and sends it ba...	
V-12	High score allows assumptions on health state.	
V-13	Analytics data includes detailed blood-sugar values.	
V-14	User sends device to company for repair without dele...	
V-15	Analytics data includes detailed location data.	
V-16	User sends device to company for repair without dele...	
V-17	High scores include location information.	
V-18	Long-term health information includes location data.	
V-19	Parent alert system always provides location information.	
V-20	Privacy policy has not been presented to user.	
V-21	User ignored privacy policy and did not read it.	

Figure 15: Hazardous system states can be collected in XSTAMPP under their appropriate category.


XSTAMPP -STPA for Privacy Project->Electronic Health Candy Castle->Establish Fundamentals->Linking of Adverse Consequences and Vulnerabilities
Project Explorer
Preferences
Electronic Health Candy Castle [priv]
0 Establish Fundamentals
System Description
System Goals
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and V...
Privacy Constraints
Design Requirements
High Level Control Structure Model
1 Privacy-compromising Control Actions
Control Actions
Privacy-compromising Control Actions T...
Corresponding Privacy Constraints
2 Causal Analysis
Control Structure With Process Model
Context Tables
Refined Privacy-compromising Cont...
Refined Privacy Constraints
Basic Scenarios
Causal Factors Table
LTL Table
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and Vulnerabilities
Switch to Vulnerabilities
Add
Add All
Remove
Remove Al
There are unsaved changes

Vulnerabilities

ID	Title
V-1	Urban planners can link individual data sets so they know...
V-2	Other players can estimate health state of player.
V-3	Other players can see identity (name, address) of player.
V-4	Urban planners can identify player from provided gps- an...
V-5	An insurance company gets to know about the existence...
V-6	Insurance company has access to detailed blood-sugar v...
V-7	Insurance company has access to detailed location data.
V-8	Smart device company has access to detailed blood-sug...
V-9	Smart device company has access to detailed location data.
V-10	Other players can track location of player.
V-11	Doctor receives detailed location information.
V-12	Parents can track location of children.
V-13	User is not aware of active analytics program and is ther...

Vulnerabilities available for linking

ID	Title
V-6	Submitted data sets include pattern, that can identify player.
V-7	The system reveals who is using the app.
V-8	Detailed blood-sugar values are sent to insurance compa...
V-9	User decides to stop using the device and sends it back t...
V-13	Analytics data includes detailed blood-sugar values.
V-14	User sends device to company for repair without deleting...
V-15	Analytics data includes detailed location data.
V-16	User sends device to company for repair without deleting...
V-17	High scores include location information.
V-18	Long-term health information includes location data.
V-19	Parent alert system always provides location information.
V-20	Privacy policy has not been presented to user.

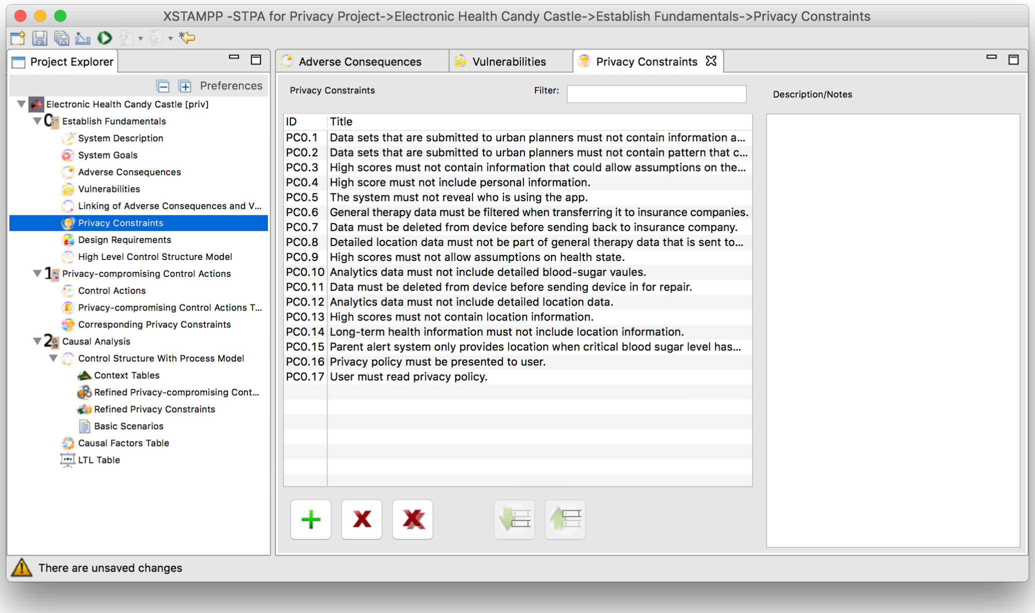
Currently linked Vulnerabilities

ID	Title
V-10	Detailed location data is sent to insurance company as pa...
V-11	User decides to stop using the device and sends it back t...
V-12	High score allows assumptions on health state.

Figure 16: Adverse consequences and their appropriate hazards can be linked in XSTAMPP, such as I did in Table 4. Select an adverse consequence from the left list and then select all hazards that are related to this adverse consequence. Click on ***Add*** to confirm.

5.2.5. Derive Privacy Constraints from Identified System Hazards

Privacy constraints are generated from the hazards list by converting the hazard into a constraint that must be valid to ensure that the hazard cannot occur. The whole list of resulting privacy constraints can be seen in Figure 17, where I used XSTAMPP in order to document the constraints. For better readability all constraints are also documented in Table 5. For instance, *Data sets submitted to urban planners include information about player* result in the privacy constraint *Data sets that are submitted to urban planners must not contain information about players*.


XSTAMPP -STPA for Privacy Project->Electronic Health Candy Castle->Establish Fundamentals->Privacy Constraints
Project Explorer
Preferences
Electronic Health Candy Castle [priv]
0 Establish Fundamentals
System Description
System Goals
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and V...
Privacy Constraints
Design Requirements
High Level Control Structure Model
1 Privacy-compromising Control Actions
Control Actions
Privacy-compromising Control Actions T...
Corresponding Privacy Constraints
2 Causal Analysis
Control Structure With Process Model
Context Tables
Refined Privacy-compromising Cont...
Refined Privacy Constraints
Basic Scenarios
Causal Factors Table
LTL Table
Adverse Consequences
Vulnerabilities
Privacy Constraints
Privacy Constraints
Filter:
Description/Notes

ID	Title
PC0.1	Data sets that are submitted to urban planners must not contain information a...
PC0.2	Data sets that are submitted to urban planners must not contain pattern that c...
PC0.3	High scores must not contain information that could allow assumptions on the...
PC0.4	High score must not include personal information.
PC0.5	The system must not reveal who is using the app.
PC0.6	General therapy data must be filtered when transferring it to insurance companies.
PC0.7	Data must be deleted from device before sending back to insurance company.
PC0.8	Detailed location data must not be part of general therapy data that is sent to...
PC0.9	High scores must not allow assumptions on health state.
PC0.10	Analytics data must not include detailed blood-sugar vaules.
PC0.11	Data must be deleted from device before sending device in for repair.
PC0.12	Analytics data must not include detailed location data.
PC0.13	High scores must not contain location information.
PC0.14	Long-term health information must not include location information.
PC0.15	Parent alert system only provides location when critical blood sugar level has...
PC0.16	Privacy policy must be presented to user.
PC0.17	User must read privacy policy.

There are unsaved changes

Figure 17: Privacy constraints are the result of analyzing possible hazardous system states.

Hazardous System States	Privacy Constraints
Data sets submitted to urban planners include information about player.	Data sets that are submitted to urban planners must not contain information about players.
Data sets submitted to urban planners include pattern that can identify individuals.	Data sets that are submitted to urban planners must not contain pattern that could help to identify individuals.
High score allows assumptions on health state.	High scores must not contain information that could allow assumptions on the health state.
High scores include personal information of player.	High score must not include personal information.
The system reveals who is using the app.	The system must not reveal who is using the app.
Detailed blood sugar values are sent to insurance company as part of the general therapy data.	General therapy data must be filtered when transferring it to insurance companies.
User decides to stop using the device and sends it back to the insurance company without deleting its content.	Data must be deleted from device before sending back to insurance company.
Detailed location data is sent to insurance company as part of the general therapy data.	Detailed location data must not be part of general therapy data that is sent to the insurance company.
High score allows assumptions on health state.	High scores must not allow assumptions on health state.
Analytics data includes detailed blood sugar values.	Analytics data must not include detailed blood sugar values.
User sends device to company for repair without deleting its content.	Data must be deleted from device before sending device in for repair.
Analytics data includes detailed location data.	Analytics data must not include detailed location data.
High scores include location information.	High scores must not contain location information.
Long-term health information includes location data.	Long-term health information must not include location information.
Parent alert system always provides location information.	Parent alert system only provides location when critical blood sugar level has been detected.
Privacy policy has not been presented to user.	Privacy policy must be presented to user.
User ignored privacy policy and did not read it.	User must read privacy policy.

Table 5: Privacy constrains derived from hazardous system states.

5.2.6. Control Structure Diagram

Control structure diagrams are a general overview of all components and subsystems of the system that should be analyzed. It reflects the hierarchical structure and layers of abstraction within the system. Interacting components are connected with an annotated arrow. This arrow represents data flow and can have different attributes which contain information about the relationship and exchanged data types.

Control structure diagrams help to identify control actions that can lead to adverse consequences and even to find additional adverse consequences. Creating the diagram itself is an iterative task and can be improved and completed during the process. In my scenario I started with a very basic control structure diagram which only contained two sensors, a smart device and a data observer. After identifying more and more participants with their relationships I have added them to the diagram with appropriate controllers.

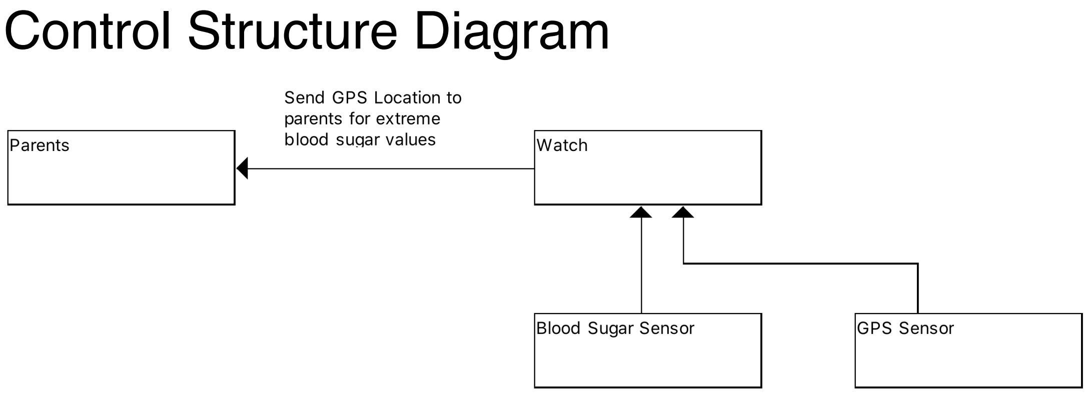
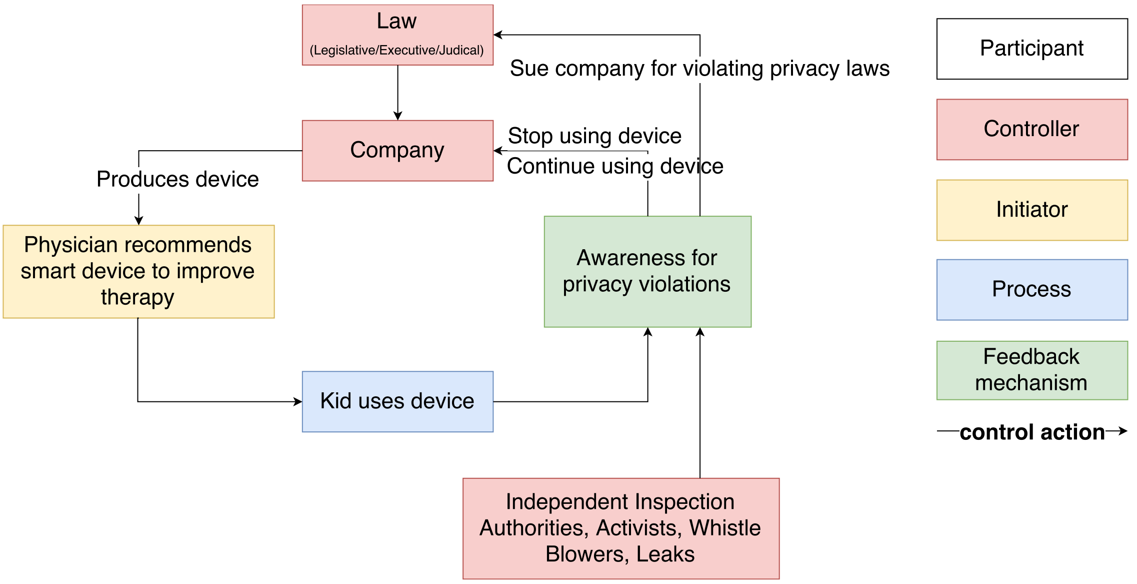

Control Structure Diagram
Send GPS Location to parents for extreme blood sugar values
Parents
Watch
Blood Sugar Sensor
GPS Sensor

Figure 18: This control structure diagram has been created using the XSTAMPP tool. This is a first draft of the control structure diagram with the most obvious elements from the scenario. In this case the smart device ("watch") is the central instance which reads blood sugar values and sends a GPS location to the parents when extreme blood sugar values occur. However, there are still many aspects missing, especially control loops have not been considered so far.

After iterating a few times on the process I decided to separate the diagram into three layers of abstraction to make them more consistent with the field of privacy; especially because the principle of feedback loops requires rethinking for privacy. The result of the first iteration can be seen in Figure 18. All components can be put together in one diagram but I split it into three parts to improve the readability.

The diagram became too complicated to be able to handle it within XSTAMPP comfortably. This is why I used a third party tool to visualize these diagrams. The first diagram in Figure 19a shows the *high-level control structure*. The company produces smart devices, which are then recommended by physicians to be used to improve the therapy of diabetes patients. The *process* is represented by the usage of this device. The awareness for privacy and privacy violations represents the *feedback mechanism* that can result in stopping using this service or device or suing the company. This awareness is the essential part of this feedback loop. The company has no motivation to consider privacy carefully without the public awareness for privacy. External individuals and organizations such as independent inspection authorities, activists, whistle blowers and leaks help understand what companies do with the patients' data and if their usage complies with laws and policies. Their recommendations help potential and current users to evaluate this product.



Law
(Legislative/Executive/Judical)
Sue company for violating privacy laws
Company
Stop using device
Continue using device
Produces device
Physician recommends smart device to improve therapy
Awareness for privacy violations
Kid uses device
Independent Inspection Authorities, Activists, Whistle Blowers, Leaks
Participant
Controller
Initiator
Process
Feedback mechanism
control action

Figure 19a: The high-level control structure of the scenario including the law, a physician, the user itself and the loop to ensure the privacy in this system.

The second control structure diagram is on user-level, as shown in Figure 19b. This represents all interaction where the user is directly involved, where the user starts an action or receives any kind of feedback. The diagram consists out of five control-loops that are all somehow connected to the user of this system. The first two are data collection controllers: The *location controller* and *blood sugar controller* ensure that the user is aware of data collection, in this case location information and blood sugar values. The next two control loops ensure that all sensitive data is removed from the device before sending it in for repair or back to the insurance company. The fifth and last user-level control loop takes care of the privacy policy that contains important information on how data is being collected, processed and shared. It is therefore very important that the user has read it and knows what happens to their data. This controller aims to ensure that the user has read the policy. This is a good example for an open-loop controller, because the controller can never ensure that the user really read the policy.

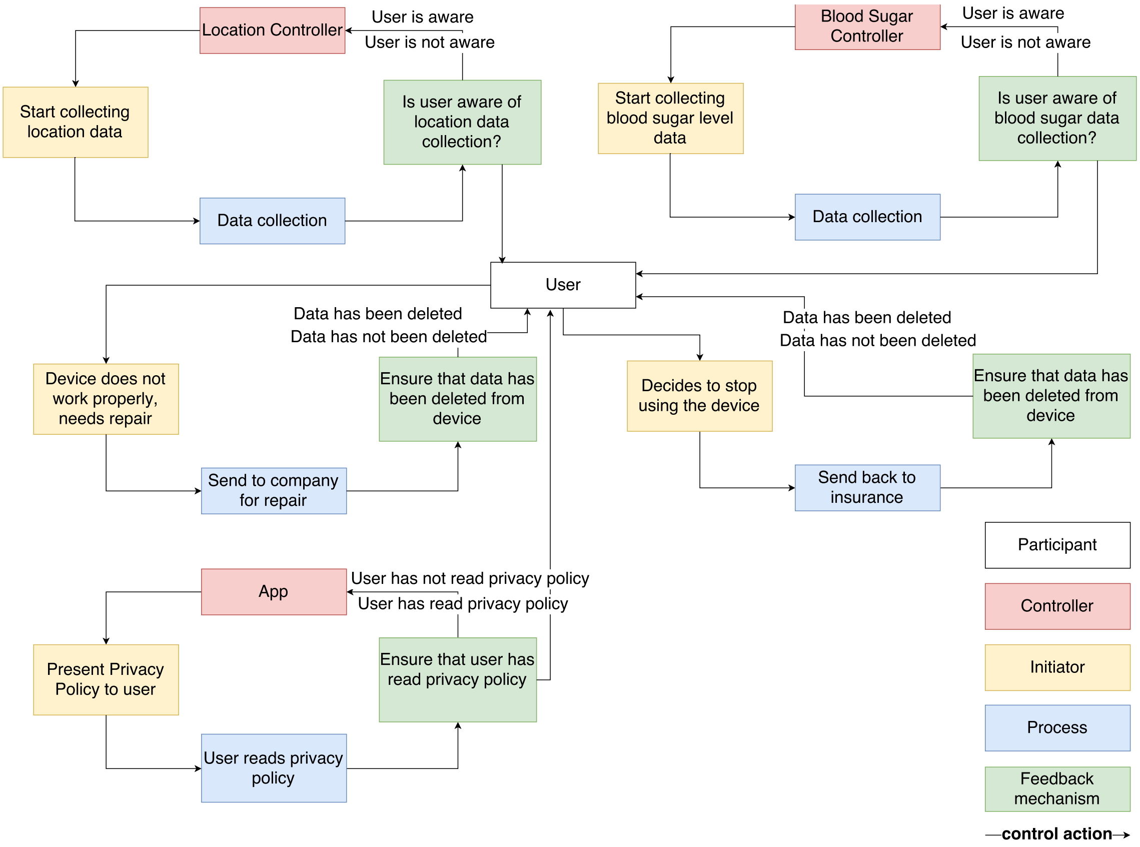

Location Controller
User is aware
User is not aware
Start collecting location data
Is user aware of location data collection?
Data collection
Blood Sugar Controller
User is aware
User is not aware
Start collecting blood sugar level data
Is user aware of blood sugar data collection?
Data collection
User
Data has been deleted
Data has not been deleted
Device does not work properly, needs repair
Ensure that data has been deleted from device
Send to company for repair
Data has been deleted
Data has not been deleted
Decides to stop using the device
Ensure that data has been deleted from device
Send back to insurance
App
User has not read privacy policy
User has read privacy policy
Present Privacy Policy to user
Ensure that user has read privacy policy
User reads privacy policy
Participant
Controller
Initiator
Process
Feedback mechanism
control action

Figure 19b: The user-level control structure diagram of the scenario.

The third part of the control structure diagram represents all technical control loops. These ensure that certain data sets do not leave the device without being filtered or are not shared with third parties at all, as shown in Figure 19c. The controllers handle data sets and ensure that different participants receive appropriately filtered data sets. The *health condition controller* collects data from the *location controller* and *blood sugar controller* and handles the filtering and dissemination of these data sets. Results are shared with the *urban planners*, *long term health controller* and *game controller*. The long term health controller provides data sets for the physician in order to analyze and improve the therapy.

The game controller itself generates data for the *high score controller* and the *usage controller*.

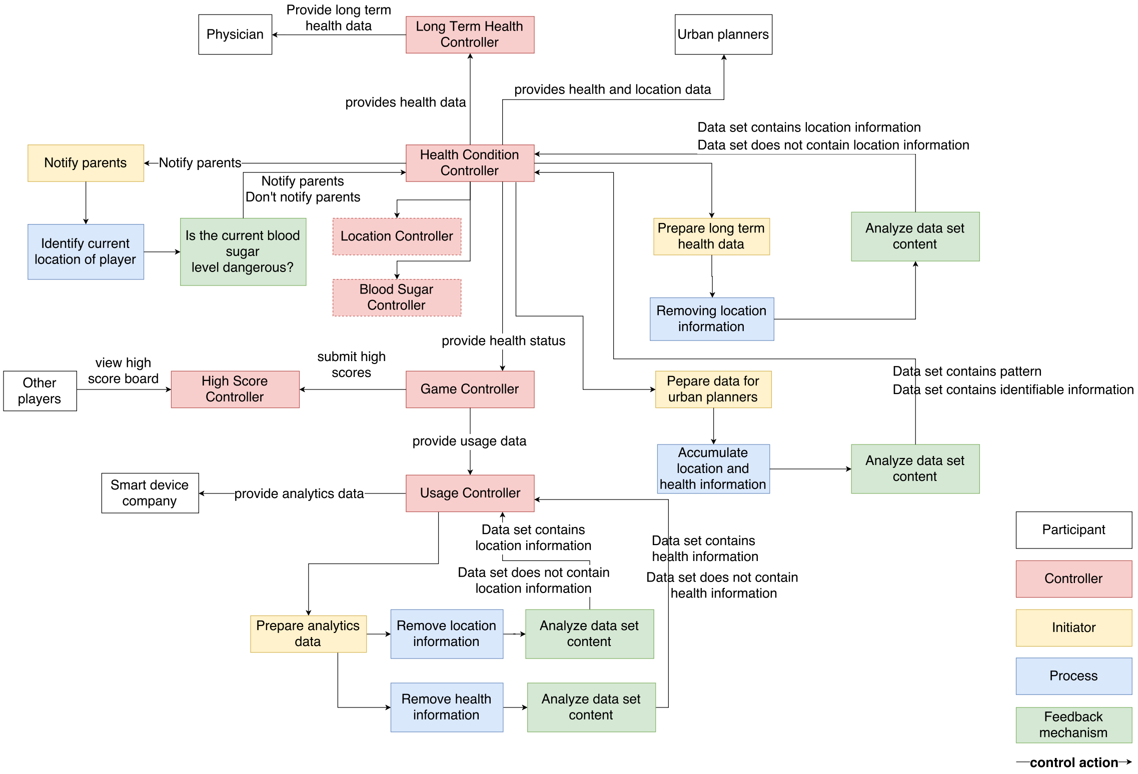

Physician
Provide long term health data
Long Term Health Controller
Urban planners
provides health data
provides health and location data
Data set contains location information
Data set does not contain location information
Notify parents
Notify parents
Health Condition Controller
Notify parents
Don't notify parents
Identify current location of player
Is the current blood sugar level dangerous?
Location Controller
Blood Sugar Controller
Prepare long term health data
Analyze data set content
Removing location information
provide health status
Other players
view high score board
High Score Controller
submit high scores
Game Controller
Pepare data for urban planners
Data set contains pattern
Data set contains identifiable information
provide usage data
Accumulate location and health information
Analyze data set content
Smart device company
provide analytics data
Usage Controller
Data set contains location information
Data set does not contain location information
Data set contains health information
Data set does not contain health information
Participant
Controller
Initiator
Process
Feedback mechanism
Prepare analytics data
Remove location information
Analyze data set content
Remove health information
Analyze data set content
control action

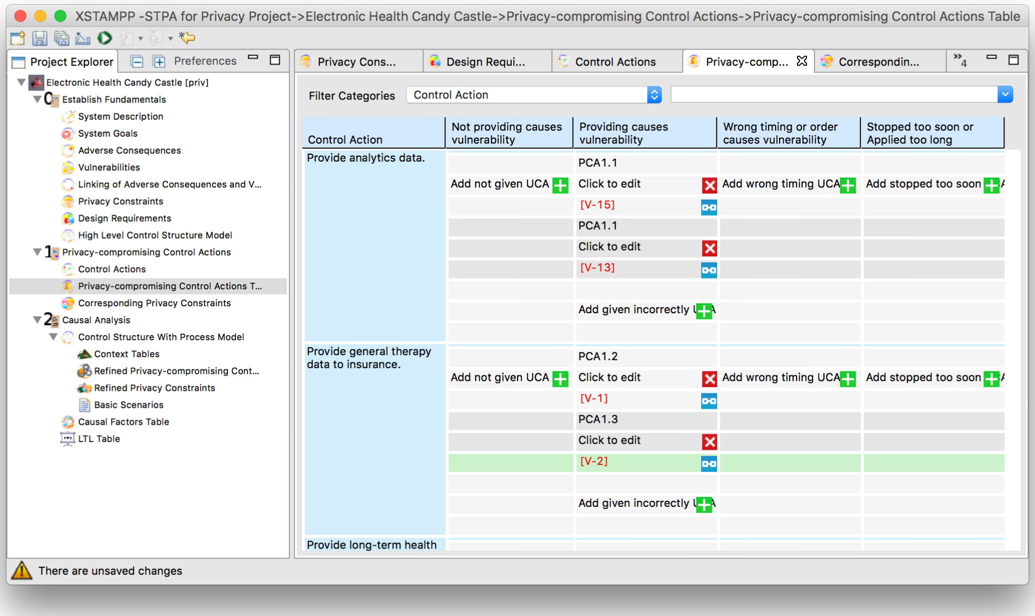
Figure 19c: The control structure diagram of all technical control loops. The *location controller* and *blood sugar controller* are references to the user-level control loops from Figure 19b.

5.2.7. Step 1: Identify Privacy-Compromising Control Actions (UCAs)

The goal of this step is to find control actions from the control structure that could violate the previously defined privacy constraints. This can occur in one of the four categories (*not providing*, *providing*, *providing too early or too late or in wrong order* or *stopping the control action too soon*). The privacy-compromising control actions together with their privacy-compromising system behavior are listed in Table 6 and in Figure 20 using XSTAMPP.

Privacy-Compromising Control Action	Caused by…	Privacy-Compromising System Behavior
Provide analytics data.	Providing…causes hazard	Sending analytics data when user is not aware of analytics program.
		Providing analytics data when data includes blood sugar information.
		Providing analytics data when data includes location information.
Provide general therapy data to insurance.	Providing…causes hazard	Providing therapy data to insurance company when data includes detailed blood sugar values.
		Providing therapy data to insurance company when data includes location information.
Provide long-term health information to physician.	Providing…causes hazard	Providing long-term health information to physician when data includes location information.
Provide health and location data for urban planners.	Providing…causes hazard	Providing health and location data when data includes information about player.
		Providing health and location data when data sets include pattern that could lead to identification.
		Providing health and location data when data sets include pattern that could help to link it to other data sets.
Reset device.	Not providing… causes hazard	Not deleting therapy data from device when sending back to insurance.
		Not deleting therapy data from device when sending to company for repair.
Submit high scores.	Providing…causes hazard	Submit high score when score could reveal health state information.
		Submit high score when score includes location information.
		Submit high score when score includes personal information.
	Wrong timing… causes hazard	Submit high score when score could reveal health state.
Send location to parents.	Providing…causes hazard	Send location to parents when no extreme blood sugar value is present.

Table 6: The *privacy-compromising control actions table* shows privacy-compromising system behavior.


Project Explorer
Preferences
Electronic Health Candy Castle [priv]
0 Establish Fundamentals
System Description
System Goals
Adverse Consequences
Vulnerabilities
Linking of Adverse Consequences and V...
Privacy Constraints
Design Requirements
High Level Control Structure Model
1 Privacy-compromising Control Actions
Control Actions
Privacy-compromising Control Actions T...
Corresponding Privacy Constraints
2 Causal Analysis
Control Structure With Process Model
Context Tables
Refined Privacy-compromising Cont...
Refined Privacy Constraints
Basic Scenarios
Causal Factors Table
LTL Table
Privacy Cons...
Design Requi...
Control Actions
Privacy-comp...
Correspondin...

XSTAMPP -STPA for Privacy Project->Electronic Health Candy Castle->Privacy-compromising Control Actions->Privacy-compromising Control Actions Table

Filter Categories: Control Action

Control Action	Not providing causes vulnerability	Providing causes vulnerability	Wrong timing or order causes vulnerability	Stopped too soon or Applied too long
Provide analytics data.	Add not given UCA ✚	PCA1.1 Click to edit ☒ [V-15] PCA1.1 Click to edit ☒ [V-13] Add given incorrectly U✚A	Add wrong timing UCA✚	Add stopped too soon ✚A
Provide general therapy data to insurance.	Add not given UCA ✚	PCA1.2 Click to edit ☒ [V-1] PCA1.3 Click to edit ☒ [V-2] Add given incorrectly U✚A	Add wrong timing UCA✚	Add stopped too soon ✚A
Provide long-term health				

There are unsaved changes

Figure 20: Control actions can be added in the appropriate section within XSTAMPP. They are then listed within the *Privacy-Compromising Control Actions Table*. This table allows to link them to specific hazardous system states.

5.2.8. Step 2: Generate Causal Scenarios

The last step of STPA-Priv concludes the privacy-risk analysis. This step generates causal scenarios that result in privacy-compromising control actions, which then violate privacy constraints. The violation of privacy constraints could then lead to a hazardous system state which could result in an adverse consequence. In other words: Adverse consequences can be prevented by eliminating causal scenarios that could trigger privacy-compromising control actions.

Table 7 lists one or more causal scenario for each privacy-compromising control action. This helps developers identify problems and fix them. For instance the privacy-compromising control action *Sending analytics data* has two causal scenarios: *The privacy policy has not been presented to the user so the user was not able to understand which data is used for analytics* and *The user did not read the privacy policy so the user does not know which data is used for analytics*.

Caused by Privacy-Compromising Control Actions	Causal Scenarios
Sending analytics data when user is not aware of analytics program.	The privacy policy has not been presented to the user so the user was not able to understand which data is used for analytics.
	The user did not read the privacy policy so the user does not know which data is used for analytics.
Providing analytics data when data includes blood sugar information.	Usage controller filters data incorrectly.
Providing analytics data when data includes location information.	
Providing therapy data to insurance company when data includes detailed blood sugar values.	
Providing therapy data to insurance company when data includes location information.	
Providing long-term health information to physician when data includes location information.	Health condition controller filtered data incorrectly.
Providing health and location data when data includes information about player.	Health condition controller filtered data incorrectly.
Providing health and location data when data sets include pattern that could lead to identification.	
Providing health and location data when data sets include pattern that could help to link it to other data sets.	
Not deleting therapy data from device when sending back to insurance.	The user does not delete their data before sending it back to insurance. The insurance can therefore access sensitive data that is stored on this device.
Not deleting therapy data from device when sending to company for repair.	The user does not delete their data before sending the device to company for repair.
Submit high score when score could reveal health state information.	The existence and frequency of submitted scores could reveal information.
Submit high score when score includes location information.	The game controller does not filter data correctly and sends location information to the score board.
Submit high score when score includes personal information.	The game controller does not filter data correctly and sends personal information to the score board.

Caused by Privacy-Compromising Control Actions	Causal Scenarios
Send location to parents when no extreme blood sugar value is present.	The health condition controller calculated the blood sugar level wrong or analyzed sensor values incorrectly. The parents are therefore notified accidentally.

Table 7: Causal scenarios for hazardous system behavior that has been caused by privacy-compromising control actions.

5.3. Applicability of STPA-Priv for Eliciting Privacy Risks in our Scenario

The application of STPA-Priv to the scenario shows that STPA's extension for privacy is indeed applicable and delivers useful results. I was able to identify participants and their relationships which helped to identify adverse consequences within the privacy risk model. Resulting hazardous system states and privacy constraints could be generated similarly to standard STPA. The resulting privacy-compromising control actions could be generated using the control structure diagrams. In the end, I was able to reveal different privacy risks in form of casual scenarios that can be used for testing and improving the system.

In addition to Shapiro's changes to the terminology [6] of STPA I had to improve the process in order to make it more consistent: I restructured the tasks in the pipeline and refined each step to make the whole process of STPA-Priv more consistent with STPA. I added a preparation step which includes identifying participants and relationships. This helps to identify adverse consequences and the draw the control structure diagrams later on. In contrast to Shapiro I used the LINDDUN privacy threat model which provides a finer granularity of privacy threats. Better results can be expected from a threat model with finer granularity in contrast to Shapiro's approach using a coarse framework. I improved the control structure diagram, which has not received much attention in Shapiro's publication. I tried different strategies to implement control structures for this scenario. In the end, I decided to split them up into three layers of abstraction in order to clarify their meaning. *High-level* and *user-level* control loops make sense and are useful for the following steps of STPA-Priv's pipeline. I renamed *actuators* into *initiators* and *sensors* into *feedback mechanisms.* Even with the renaming it is still unclear if *technical feedback loops* are really useful for privacy evaluation: Does it make sense in real software projects to ensure that data sets have been cleared from certain data points, as described in section 4.3?

5.3.1. Usability of XSTAMPP for STPA-Priv

To support the analysis process I used the existing software *XSTAMPP* which has been developed by University of Stuttgart [21, 22]. XSTAMPP offers an implementation for the process of STPA, which is useful for STPA-Priv as well. There has already been an update for XSTAMPP which supports STPA-Priv that I described in sections 2.4 and 4.2.

Several changes have already been implemented in XSTAMPP in order to support STPA-Priv: *Losses* are labeled *Adverse Consequences, Security* or *Safety Constraints* are labeled *Privacy Constraints* and *Insecure* or *Unsafe Control Actions* are labeled *Privacy-Compromising Control Actions.*

These changes make sense and help engineers to elicit privacy risks. However, some modifications in the software are still necessary to be consistent with STPA-Priv and improve the overall experience. Larger control structure diagrams in XSTAMPP are not well-arranged. I had to use a third-party tool in order to provide an appropriate representation for this bachelor's thesis. The visual editor within XSTAMPP should be improved to handle larger control structures as well.

It would be useful to represent the three layers of abstraction for control loops in XSTAMPP as well. It would be great if they can be created in three different views, but can still be linked to each other. If linked correctly they could be shown in an overall diagram together or in three separate diagrams to improve the readability.

Vulnerabilities should be renamed to *hazardous system states*.

In the view *Linking Adverse Consequences and Vulnerabilities* there is a wrong label: The table with adverse consequences is titled vulnerabilities. This is wrong. They should be titled adverse consequences instead.

After restarting the computer, the STPA-Priv extension for XSTAMPP disappeared frequently and had to be reinstalled.

After all, XSTAMPP is already useful for the use of STPA-Priv. The additional suggestions to improve the software are not requirements that make the software unusable right now. However, they would increase the usability. Especially the control structure diagram editor within XSTAMPP required using a third party tool in order to create better readable diagrams.

6. Conclusion and Future Work

When analyzing systems for privacy risks it is useful to use an existing process that helps to elicit risks, such as STPA-Priv. I described the extension of STPA for privacy, STPA-Priv, and how it can be used for privacy engineering. I described a realistic eHealth scenario that requires privacy risk analysis and used STPA-Priv to elicit privacy risks in this scenario. I was able to generate adverse consequences, a control structure diagram of my scenario, privacy-compromising control actions and, in the end, scenarios in which privacy is at risk. This shows that the general idea of STPA-Priv works and that it is helpful in real-world scenarios. The straightforward and well-defined methodology which is based on STPA is not limited to single components but considers all layers of abstraction within systems.

The application of STPA-Priv to the eHealth scenario shows that disadvantages of existing privacy analysis techniques can be overcome by STPA-Priv. This includes the ability to cope with various privacy requirements of complex socio-technical systems. However, it makes sense to make use of existing technologies within the pipeline of STPA-Priv: The series of open questions from section 3.3 stimulate engineers to find data flow, relationships and participants which helps to identify adverse consequences. The privacy threat catalogue from the LINDDUN methodology from section 2.6 is a great framework for privacy risks and helps identify adverse consequences as well. The control and feedback approach described in section 3.4 is one important aspect of user-level control loops, which are part of the control structure. So, after all, STPA-Priv has been able to overcome many of the problems that existing techniques have but it makes sense to continue using some of them as part of STPA-Priv.

The application of STPA-Priv to this scenario can only serve as an initiation to analyze further application scenarios with it. An empirical evaluation similar to Wuyts, Scandariato and Joosen [33] would be useful to be able to classify the effectivity of STPA-Priv. Different participants receive the same scenario, and one group should analyze it using STPA-Priv. These results are then compared to results of privacy experts in order to see how many privacy risks could be elicited using STPA-Priv. This would reveal different properties of STPA-Priv, such as correctness of risks, completeness of risks and reliability.

In any way, I can recommend using STPA-Priv to evaluate projects for privacy risks despite the fact that there are still changes needed. Nevertheless, these changes do not change the overall method which is very straight-forward and applicable to privacy.

7. Bibliography

1. M. Hilbert and P. López, The World's Technological Capacity to Store, Communicate, and Compute Information. science, 2011. p. 60-65. https://doi.org/10.1126/science.1200970

2. O. Tene, What Google knows: Privacy and Internet Search Engines. 2007. https://doi.org/10.2139/ssrn.1021490

3. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future generation computer systems, 2013. p. 1645-1660.

4. J. Gehrke, E. Lui, and R. Pass, *Towards privacy for social networks: A zero-knowledge based definition of privacy.* Theory of Cryptography Conference, 2011. p. 432-449. https://doi.org/10.1007/978-3-642-19571-6_26

5. C. Gans-Combe, Data Protection and Privacy Ethical Guidelines (Version 5). European Commission, Sept, 2009.

6. S.S. Shapiro, Privacy Risk Analysis Based on System Control Structures: Adapting System-Theoretic Process Analysis for Privacy Engineering. Security and Privacy Workshops (SPW), 2016 IEEE, 2016. p. 17-24.

7. Merriam-Webster's Collegiate Dictionary.

8. N.G. Leverson, Engineering a safer world: Systems thinking applied to safety. 2011: MIT press. 978-0-262-01662-9

9. L.v. Bertalanffy, *General System Theory.* New York, 1968. p. 40.

10. R. Stichweh, *Systems Theory.* 2011.

11. S. Ramo and R.K. St.Clair, *The systems approach.*

12. C.A. Ericson, *Hazard analysis techniques for system safety.* John Wiley & Sons, 2015. p. 365-381. https://doi.org/10.1002/0471739421

13. W.-S. Lee, D.L. Grosh, F.A. Tillman, and C.H. Lie, *Fault Tree Analysis, Methods, and Applications – A Review.* IEEE transactions on reliability, 1985. p. 194-203. https://doi.org/10.1109/TR.1985.5222114

14. M. Ben-Daya, D. Ait-Kadi, S.O. Duffuaa, J. Knezevic, and A. Raouf, Failure mode and effect analysis. Handbook of maintenance management and engineering. 2009. https://doi.org/10.1007/978-1-84882-472-0

15. C.A. Ericson, *Hazard and Operability Analysis.* Hazard Analysis Techniques for System Safety, 2005. p. 365-381.

16. A. Abdulkhaleq, S. Wagner, and N. Leveson, A comprehensive safety engineering approach for software- intensive systems based on STPA. Procedia Engineering, 2015. p. 2-11. https://doi.org/10.1016/j.proeng.2015.11.498

17. C. Harrison Fleming, M. Spencer, N. Leveson, and C. Wilkinson, *Safety Assurance in NextGen.* NASA Technical Report NASA/CR-2012-217553, 2012. https://doi.org/10.1016/j.ssci.2012.12.005

18. C.f.C.P. Safety, *Guidelines for Design Solutionsfor Process Equipment Failures.* American Institute of Chemical Engineers, 1998. p. 179 - 201.

19. S. Shapiro, From STPA*-Sec to STPA- Priv: Leveraging STPA for Privacy Engineering. 2016.

20. W. Young and N.G. Leveson, An integrated approach to safety and security based on systems theory. Commun. ACM, 2014.(2): p. 31-35. https://doi.org/10.1145/2556938

21. XSTAMPP: An eXtensible STAMP Platform As Tool Support for Safety Engineering.

22. L. Balzer, A. Abdulkhaleq, and S. Wagner. *http://www.xstampp.de*. 2012.

23. D.P. Siewiorek, Generation Smartphone – The smartphone's role as constant companion, helper, coach, and guardian has only just begun. IEEE Spectrum • September 2012, 2012. p. pages 54-58.

24. P. Mullan, C. Kanzler, B. Lorch, L. Schroeder, L. Winkler, L. Laich, F. Riedel, R. Richer, C. Luckner, H. Leutheuser, B. Eskofier, and C. Pasluosta, *Unobtrusive Heart Rate Estimation during Physical Exercise using Photoplethysmographic and Acceleration Data.* 2015. https://doi.org/10.1109/EMBC.2015.7319787

25. M. Knöll, "On the Top of High Towers..." Discussing Locations in a Mobile Health Game for Diabetics. Revista EducaOnline, 2011. p. 1-16.

26. S. Avancha, A. Baxi, and D. Kotz, *Privacy in mobile technology for personal healthcare.* ACM Computing Surveys (CSUR), 2012. https://doi.org/10.1145/0000000.0000000

27. Y. Bai, L. Dai, and J. Li, *Issues and challenges in securing eHealth systems.* International Journal of E-Health and Medical Communications (IJEHMC), 2014. p. 1-19. https://doi.org/10.4018/ijehmc.2014010101

28. J.H. Saltzer and M.D. Schroeder, *The protection of information in computer systems.* Proceedings of the IEEE, 1975. p. 1278-1308. https://doi.org/10.1109/PROC.1975.9939

29. J.I. Hong, J.D. Ng, S. Lederer, and J.A. Landay, Privacy risk models for designing privacy-sensitive ubiquitous computing systems. 2004.

30. F.o.E.d.a.i. marketing, European code of practice for the use of personal data in direct marketing electronic communications annex. 2010.

31. R. Calo, *The boundaries of privacy harm.* 2010. p. 1131-1162. https://doi.org/10.2139/ssrn.1641487

32. K. Wuyts, R. Scandariato, and W. Joosen, *LIND (D) UN privacy threat tree catalog.* 2014.

33. K. Wuyts, R. Scandariato, and W. Joosen, *Empirical evaluation of a privacy-focused threat modeling methodology.* Journal of Systems and Software, 2014. p. 122-138. https://doi.org/10.1016/j.jss.2014.05.075

34. S. Brooks, E. Nadeau, M. Garcia, N. Lefkovitz, and S. Lightman, *Privacy Risk Management for Federal Information Systems.* National Institute of Standards and Technology, U.S. Department of Commerce, 2015.

35. K. Lu, Z. Li, V.P. Kemerlis, Z. Wu, L. Lu, C. Zheng, Z. Qian, W. Lee, and G. Jiang, Checking More and Alerting Less: Detecting Privacy Leakages via Enhanced Data-flow Analysis and Peer Voting. 2015. https://doi.org/10.14722/ndss.2015.23287

36. M. Egele, C. Kruegel, E. Kirda, and G. Vigna, *PiOS: Detecting Privacy Leaks in iOS Applications.* 2011. p. 177-183.

37. W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L.P. Cox, J. Jung, P. McDaniel, and A.N. Sheth, *TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones.* ACM Transactions on Computer Systems (TOCS), 2014. https://doi.org/10.1145/2494522

38. A.S. Victoria Bellotti, Design for Privacy in Ubiquitous Computing Environments. 1993. https://doi.org/10.1017/CBO9781107415324.004

39. M. Meingast, T. Roosta, and S. Sastry, *Security and privacy issues with health care information technology.* Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE, 2006. p. 5453-5458.

40. A. Kaletsch and A. Sunyaev, Privacy Engineering: Personal health records in cloud computing environments. 2011. 9781618394729

41. @AbdulkhaleqAsim. *Tweet*. 2017; Available from: https://twitter.com/AbdulkhaleqAsim/status/837667755868434435.

42. C. Stach, *Secure Candy Castle – A Prototype for Privacy-Aware mHealth Apps.* Mobile Data Management (MDM), 2016 17th IEEE International Conference on, 2016. p. 361-364. https://doi.org/10.1109/MDM.2016.64

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift