Exploratory Study of the Privacy Extension for System Theoretic Process Analysis (STPA-Priv) to elicit Privacy Risks in eHealth

Kai Mindermann, Frederik Riedel, Asim Abdulkhaleq, Christoph Stach, Stefan Wagner
University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart

***Abstract*—*Context:* System Theoretic Process Analysis for Privacy (STPA-Priv) is a novel privacy risk elicitation method using a top down approach. It has not gotten very much attention but may offer a convenient structured approach and generation of additional artifacts compared to other methods. *Aim:* The aim of this exploratory study is to find out what benefits the privacy risk elicitation method STPA-Priv has and to explain how the method can be used. *Method:* Therefore we apply STPA-Priv to a real world electronic health scenario that involves a smart glucose measurement device used by children. Different kinds of data from the smart device including location data should be shared with the parents, physicians and urban planners. This makes it a sociotechnical system that offers enough and complex privacy risks to be found. *Results:* We find out that STPA-Priv is a structured method for privacy analysis and finds complex privacy risks. The method is supported by a tool called XSTAMPP which makes the analysis and its results more profound. Additionally, we learn that an iterative application of the steps might be necessary to find more privacy risks when more information about the system is available later. *Conclusions:* STPA-Priv helps to identify complex privacy risks that stem from sociotechnical interactions in a system. It also outputs privacy constraints that are to be enforced by the system to ensure privacy.**

I. Introduction

The increasing importance of privacy is relevant for organizations and individuals in a connected world. Especially for upcoming electronic health systems that are enabled by Internet of Things devices. Such complex socio-technical software systems offer personalized services, personal assistants and cloud services. Collecting and processing personal information is essential for those services. According to a Gartner forecast [1] the Internet of Things, as in everything from your toothbrush over your freezer to your television is connected to the Internet, will encompass 26 billion devices by 2020. Each of these devices will have different sensors ranging from a camera, microphone and GPS sensors to more unfamiliar ones like motion, temperature and light sensors (and even more specific ones). As multiple devices have these sensors and they send the collected data usually to at least one service provider it is unknown on what basis these data is analyzed and related to other information. The research project PATRON (Privacy in Stream Processing) funded by the Baden-Württemberg Stiftung works on concealing privacy relevant patterns in data streams. It is important to find relevant privacy vulnerabilities to derive patterns. Previous analysis techniques appear to be non-systematic when it comes to the combination of data and interaction with the environment. For example with multiple data receivers that are not part of the original system. Safety and security can be considered a system property and system-theoretic approaches are being used as alternatives to established methods in their field. Privacy can also be seen as a system property as privacy relevant data can be leaked at all abstraction levels. Therefore we expect that system-theoretic methods can be applied with good results to privacy as well. The currently proposed STPA-Priv method by Shapiro [2] has not been getting a lot of attention and we want to explore if it is feasible to use the method for privacy analysis.

II. Related Work

Data-flow analysis techniques have been developed to track data flow and elicit privacy risks. The approach described by Lu and Li [3] 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 Krueger [4]. Their system is able to detect data-flow in compiled Objective-C binaries, similar to Lu and Li's approach. Another interesting approach has been developed by Enck and Gilbert [5]. Their system can analyze data-flow in Android applications in real-time, in contrast to the static approach of Lu, Li, Egele and Krueger. Their system TaintDroid can be run on productive devices in the background to spot malicious app requests.

Analyzing data flow, such as suggested by Lu and Li [3], Egele and Krueger [4] and Enck and Gilbert [5], 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 these information outside of their scope. In many cases, it is necessary to exchange information for a service to be able to work as expected. The revealing of privacy 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 [6]. Their 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*, *unwareness* and *non-compliance* (hence the name LINDDUN). The model goes even further and describes a process to resolve privacy threats. 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 is a drawback of this approach that has been proven by Wuyts, Scandariato and Joosen in a set of extensive empirical studies [7]. They also state that it "[...] mainly focuses on the privacy of the data subject (i.e. the person the data are about). Rather than focusing on internal processes and flows[...]" [8]. This is where STPA-Priv could be useful with its top-down approach.

III. STPA-Priv

To understand what STPA-Priv is we first describe its origin and what ideas constituted to the development of STPA. Then we introduce the STPA-Priv extension and available tool support.

A. STAMP and STPA

Leveson developed a new accident model based on system and control theory called STAMP (Systems-Theoretic Accident Modeling and Processes) [9]. In STAMP, accidents are considered results from inadequate enforcement of safety constraints in system design, development and operations. STAMP treats safety as control problem rather than component failures. In STAMP, the system is seen as a set of control components which interact with each other. This helps to create models of systems which cover human, technology, software, and environmental factors, such as governmental policy [9]. Therefore, STAMP considers accidents not only arising from individual component failures but also from the interaction among system components. In other words, accidents occur when component failures, external disturbances and/or dysfunctional interactions among system components are not adequately handled by the safety control system [9]. Based on STAMP, a new method for hazard analysis called STPA (System-Theoretic Process Analysis) was developed to identify hazards existing in the system and providing so-called safety constraints to mitigate those hazards.

STPA has been extended to support the security analysis based on systems theory. Young and Leveson [10] developed an approach called STPA-Sec (STPA for security) which extends the STPA safety analysis with security aspects.

B. STPA-Priv Extension

Shapiro extended STPA-Sec to be used for the elicitation of privacy risks [2]. 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. The steps of STPA are in principle the same in STPA-Priv, only their terminology was changed. Losses or accidents in traditional STPA are always related to a loss of human life, injuries or destruction of expensive hardware. However, privacy violations primarily do not lead to accidents which threaten human life[1], but lead to embarrassing, awkward and adverse situations, or emotional damage in general, for individuals. This is why losses and accidents are renamed to *adverse consequences* in respect to privacy. An important property of STPA-Priv is that it can cope with open-loop controls, that is controls that are not able to provide feedback to their controlling entity (e. g. privacy policies are there but can not always be known if the user really read them [11]). An overview of the needed steps and comparison between STPA-Priv, STPA-Sec and STPA is depicted in Table I.

Wuyts and Joosen define four different knowledge classes for privacy research [12]. They are *methodology/process*, *Principle*, *Guideline* and *Pattern*. Currently we would classify STPA-Priv as a *methodology/process*. But as we will learn in this exploratory study, it has to be augmented by at least a threat catalog for a useful analysis so it might also belong to the *pattern* category.

C. Tool Support with XSTAMPP

XSTAMPP (eXtensible STAMP platform) [13] is an open-source platform for safety engineering developed specially to support the STAMP methodologies. XSTAMPP is written in Java based on the Eclipse Plug-in-Development Environment (PDE) and Rich Client Platform (RCP). XSTAMPP support safety engineers to perform both safety and security analysis based on STPA. XSTAMPP also supports the software engineers to perform safety-based testing and verification activities based on the STPA safety analysis results. The current version of XSTAMPP[2] 2.1.1 supports the safety, security and privacy analysis based on STPA.

IV. Scenario Description (eHealth)

For the exploratory application of STPA-Priv we need a reasonable scenario which we describe in this section.

Since chronic diseases such as diabetes mellitus are on the rise, the healthcare system has to face high treatment costs and overburdened physicians. As a consequence, novel treatment methods are badly needed. eHealth, i. e., the usage of common computing systems such as PCs or smartphones for health care, is such a method. With eHealth the patients are able to perform periodic screenings at home instructed only by their computing system. eHealth applications can even be tailored to almost any

[1]Privacy violations can still lead to political or other kinds of persecution and should therefore not be neglected.

[2]www.xstampp.de

	STPA-Priv	STPA-Sec	STPA
Step 0 Fundamentals	Define System Goals and Description		
	Define **Adverse Consequences**	Define Losses	Define Accidents
	Define Vulnerabilities		Define Hazards
	Link Adverse Consequences to Vulnerabilities	Link Losses to Vulnerabilities	Link Accidents to Hazards
	Specify **Privacy Constraints**	Specify Security Constraints	Specify Safety Constraints
	Specify Design Requirements		
	Create Control Structure Model		
Step 1 Control Actions	Derive Control Actions		
	Define **Privacy-compromising Control Actions**	Define Unsecure Control Actions	Define Unsafe Control Actions
	Corresponding Privacy Constraints	Corresponding Security Constraints	Corresponding Safety Constraints
Step 2 Causal Analysis	Derive Causal Factors		

Table I
OVERVIEW OF THE STEPS OF THE STPA-PRIV METHOD WITH COMPARISON TO THE ORIGINAL STPA AND STPA-SEC.

given medical condition [14]. However, eHealth is not just able to reduce treatment costs. With the help of *serious games* it is possible to integrate required therapeutic procedures into the daily routines of child patients. That way, the young patients are able to cope with their illness much better [15].

Knöll develops in cooperation with the Olgahospital Stuttgart, a children's hospital in Germany, an idea for a serious game for children suffering from diabetes [16]. In this smartphone game the player, i. e., the patient, has to enter his or her blood sugar values regularly. Each of these entries is enriched with the current location of the player and a timestamp[3]. The game somehow has to motivate the player not only to check his or her blood sugar level regularly but also to do this at varying locations.

From the player's point of view, this concept is beneficial as s/he has to monitor the blood sugar level anyhow. Accordingly the game is both a motivation as well as a reminder. Additionally physicians benefit from such a game. Normally patients have to keep a handwritten diabetes diary and the physicians have to review the diary. Yet, these diaries contain wrong or incomplete data and they are difficult to decipher. The game-based approach is able to create an electronic diabetes diary without the drawbacks of the paper-based version.

However, there are even more stakeholders for such a game. Due to the augmented health data, also urban planners can profit from health games. With the help of physicians, they are able to identify unhealthy places in town, i. e., places which have a bad influence on the patient's health. Based on this knowledge, correlations between architectural characteristics (e. g., crowded streets) and health condition changes can be deduced [17].

It is obvious that not every involved party requires all of the captured data especially since this kind of data is highly sensitive data from a privacy point of view. Therefore it is strongly recommended to restrict the party's data access. E. g., the urban planners need only to know which locations have an influence on the health condition. However, they need no access to any actual health data or even data from which they are able to draw inferences about the patient. The physicians on the other hand need access to the whereabouts of a patient only in case of an emergency. Such a scenario requires fine-granular privacy mechanisms which assure the best quality of service and conceal as much private data as possible. The studies mentioned above do not take privacy into consideration. The participants had to agree to the unrestricted usage of their data and had to trust in the good will of the involved parties.

[3]Both, the location as well as the current time can be captured by the sensors built in the smartphone.

V. APPLICATION OF STPA-PRIV

In the following we want to systematically analyze this scenario using STPA-Priv for any privacy risks the involved parties have to be aware of and how these risks can be mitigated or even prevented. The analysis follows the steps listed in Table I.

Step 0: Fundamentals

1) Define System Goals and Description: Important goals of the system have to be kept in mind in all further steps, as the system must always fulfill its goal. We can get important information from the system description, such as the involved parties that share or process data. In this scenario we come up with the following initial involved parties: *Child* (as in the person that has diabetes), *parents*, *physician*, *insurance company*, *smart device manufacturer* and *other players* of the game (other children with diabetes).

2) Define Adverse Consequences: Finding and defining **adverse consequences** in our system is an important step of STPA-Priv and requires experts that know the scenario and its entities. However, it is not necessary to know the implementation of each component, since STPA is a top-down approach and works at the system and component level.

This step can and should be augmented by a systematic catalog of privacy threats, such as *LINDDUN privacy threat tree*

catalog [6] or *Calo's subjective/objective privacy harms* [18] (as Shapiro used in his initial proposal of STPA-Priv [2]). 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 [7]. Its threat trees have been considered useful in practice. It provides privacy analysis methods as well, however, we only utilized their threat tree in our case. It offers privacy threats from different categories: *linkability*, *identifiability*, *non-repudiation*, *detectability*, *information disclosure*, *unwareness* and *non-compliance*. Threats from these categories are then used to find adverse consequences.

Each adverse consequence can be triggered by one or more system states together with environmental conditions of the system. These are called **vulnerabilities or vulnerable system states**. Vulnerabilities are system states that are under the system's control, whereas adverse consequences themselves are not controllable. This is why vulnerable system states have to be prevented. Elaborating adverse consequences is the counterpart to accidents in original STPA. Here we can use the knowledge from the previous step about the involved parties and their relationships.

As an example, the relationship between the child and the smart device manufacturer is of commercial interest. Exchanged data includes analytics data and crash report information. Applying different privacy threats from LINDDUN threat tree create the following adverse consequence: *The user is not aware of active analytics program and is therefore suspect to surveillance*. This is a result of the general privacy threat *unawareness*. Another example is *other players can estimate health state of player* which is caused by *information disclosure* [2]. More adverse consequences are listed in Table II.

3) Define Vulnerabilities: Now that we have a list of adverse consequences we need to define corresponding **vulnerable system states** that can lead to these adverse consequences [2] [9]. Depending on the adverse consequence we can define an abstract system state description for each adverse consequence that would be exploitable, respectively that can lead to the adverse consequence. At this point we have no list or model of possible system states. Therefore, we assume we have to describe them in a textual form with our domain knowledge and the knowledge of the system and they do not have to be actual states in the implemented components of the system.

The adverse consequence *the user is not aware of active analytics program and is therefore suspect to surveillance* can be caused by the system states *privacy policy has not been presented to user* and *user ignored privacy policy and did not read it*. A list of identified vulnerable system states of our scenario is listed in Table II where we also state the adverse consequence and the LINDDUN category.

4) Link Adverse Consequences to Vulnerabilities: As already inferable from the Table II the vulnerable system states should be linked to all the adverse consequences that can be caused by them. This has to be done iteratively for each adverse consequence.

5) Specify Privacy Constraints: **Privacy constraints** ensure that vulnerable system states do not occur. They are created basically by negation of the vulnerabilities. As an example, the vulnerability *General therapy data includes detailed blood sugar values* can be converted to the privacy constraint *Exported therapy information must not include detailed blood sugar values*. (Here we should be more specific what *detailed* means, but for this exploration it should be enough).

If we make sure all the privacy constraints are enforced/followed correctly then the previously defined adverse consequences are prevented. This is why we want to find control actions that could violate these constraints in the following steps.

6) Specify Design Requirements: This step is skipped for this exploration.

7) Create Control Structure Model: An important part of STPA is the System **Control Structure Model**. It contains the *processes* that are to be controlled using a so called *feedback-loop*. The feedback loop is created by attaching a *sensor* that checks the process and reports to a *controller*. The controller then evaluates the sensor value and uses an *actuator* to control the process again. The terminology originates from the safety anaylsis where the system is build up of these parts. It is not yet defined how these elements have to be used in the privacy analysis. When analyzing existing systems we can eventually use their existing control structure diagram in this step. If a new system is analyzed we have to create a control structure diagram.

The control structure diagram of our scenario is depicted in Figure 1. There are many involved parties with different relationships and interests: The *child* plays an important role; it uses the *smart device*. The smart device is capable of *measuring the blood sugar level* and can *locate its position* using the Global Positioning System (GPS). Whenever a blood sugar change occurs, the *blood sugar controller* is triggered. This controller decides whether an action is necessary: It could notify the *game controller* to motivate the patient to inject insulin in exchange for in-game rewards, and it can notify the *parent alert controller* in case of an extreme blood sugar value to get help for the child. The game controller also includes a *high score controller* which can *share scores* with other players of the game. *Physicians* can access long-time measurements to be able to discuss and improve the therapy. Analytics data, usage data and crash reports of the smart device are sent out to the *smart device manufacturer* to improve their service or the device. The *health insurance company* is interested in general usage data to be able to see if participants are using the smart device on a regularly basis and correctly. Using this technology more often leads to better insulin injection results and a more stable health condition of the child. This decreases the expenses of the insurance for a specific patient and can therefore decrease their insurance contribution. The creation of this control structure is not easy and requires a lot of domain knowledge. Also it requires decisions on where the system boundaries are and how detailed components are modeled.

Adverse Privacy Consequences	LINDDUN Category	Vulnerable System States
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.
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.
Other players can track location of player.	Information Disclosure	High scores include location information.
Other players can estimate health state of player.	Information Disclosure, Linkability	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.
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.
Urban planners can identify player from provided gps- and health-data.	Identifiability	Submitted data sets include information about player. Submitted data sets include pattern, that can identify player.
Urban planners can link individual data sets so they know that they come from the same player.	Linkability	Submitted data sets include information about player. Submitted data sets include pattern, that can identify individuals.

Table II
ADVERSE CONSEQUENCES AND THEIR VULNERABILITIES IN OUR EHEALTH SCENARIO. THEIR LINDDUN CATEGORY SHOWS BY WHICH KIND OF PRIVACY THREAT THEY ARE CAUSED.

Step 1: Control Actions

1) Derive Control Actions: The created control structure diagram is used as a basis to derive the existing control actions. They can be taken directly from the model.

2) Define Privacy-Compromising Control Actions and 3) Specify Corresponding Privacy Constraints: More important are the **privacy-compromising control actions** which violate privacy constraints when being executed. The goal of this step is to find all privacy-compromising control actions. In the end, these privacy-compromising control actions are the flaw of our system and need to be tamed by the engineers. Each privacy constraint is enforced by a controlling component (see Figure 1). According to STPA-Priv, privacy-compromising control actions can be classified in one of the following four categories: 1) Not providing the control action, when it should be provided. 2) Providing the control action, when it should not be provided. 3) Providing the control action too early, too late or in wrong order. 4) Stopping the control action or applying it too long. Yet we feel unsure whether these four categories apply for privacy. Still, it depends on how the control actions are modeled and/or actually executed.

When looking at the privacy constraints, we have to find the appropriate control actions from the control structure in Figure 1 that is responsible for ensuring the privacy constraint. As an example, the control action *send analytics data* is responsible for ensuring that *the privacy policy has been presented to the user*, that *the user must read the privacy policy*, that *analytics data must not include location information* and that *analytics data must not include blood sugar values*. These cases are then listed in their appropriate category, caused by the privacy-compromising control action *send analytics data*. This results in vulnerabilities like *sending analytics data causes vulnerability when user is not aware of analytics program* or *sending analytics data causes vulnerability when data includes blood sugar information*.

Step 2: Causal Analysis - Derivation of Causal Factors

The previous step generated a list of privacy-compromising control actions that can violate privacy constraints and therefore potentially cause vulnerable 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. 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*. We have to look at vulnerabilities that are caused by these control actions to find causal scenarios. For example he control action *Send analytics data* can cause different vulnerable states, such as *sending analytics data when user is not aware of analytics program*. This can happen when *the user did not read the privacy policy, or the agreement has not been made available to the user*. These are then the first causal scenarios. The next vulnerability is *providing analytics data when data includes detailed blood*

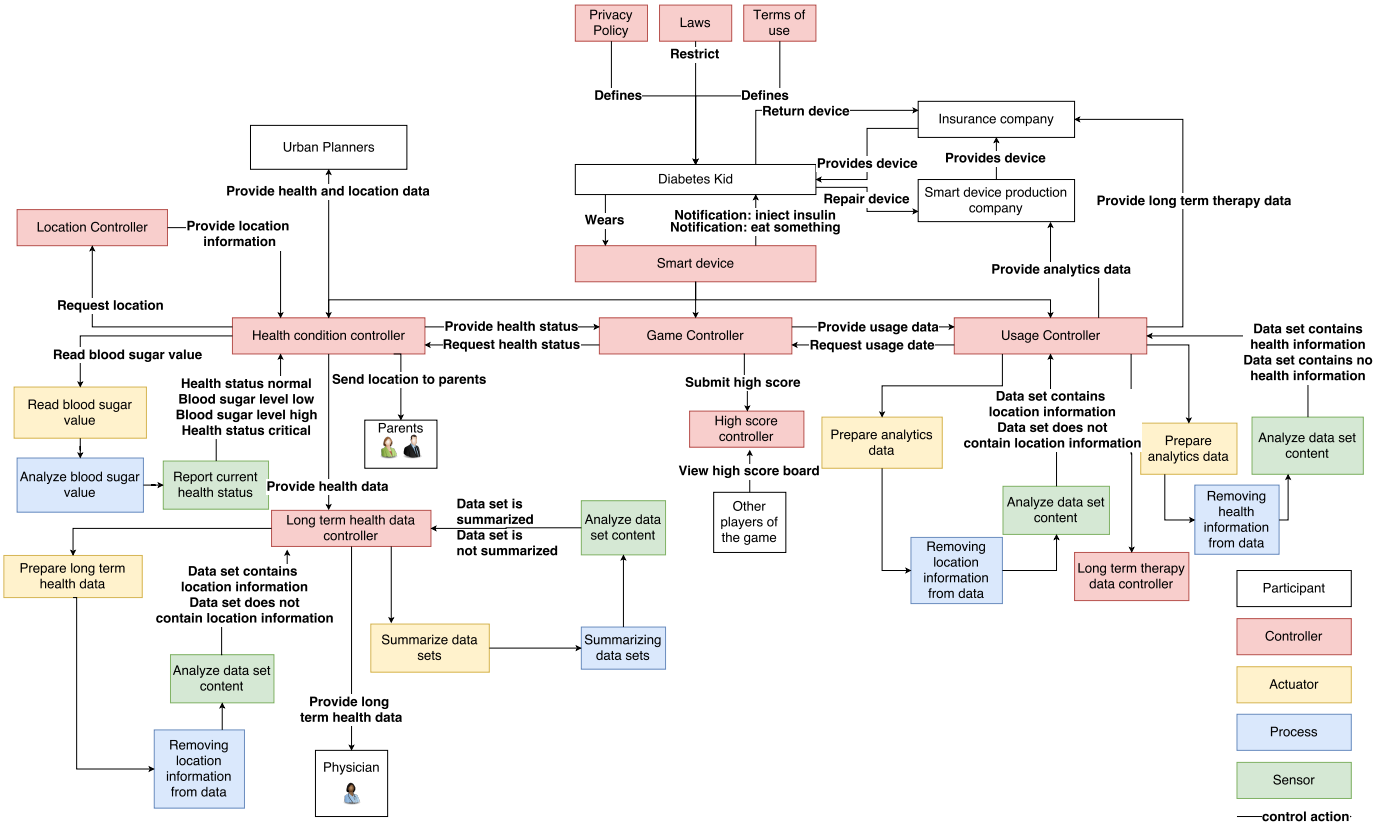

Figure 1. The system with its control structure in a diagram after several iterations. The patient itself, their parents, the smart device producing company, the physician, the insurance company and other players of the game. Different controllers within the watch ensure that data sets are only redirected to specific participants if appropriate requirements are met.

sugar values. This can be caused by a scenario in which *the usage controller filters data incorrectly*.

Control actions that can be referred to a causal scenario require a risk management response. This can be a privacy policy or terms of use that all parties and components must comply to or other kinds of actions. These are not part of the analysis method and are of course system specific.

VI. Discussion and Conclusion

This was a very brief description of the application of STPA-Priv to a complex eHealth scenario.

The application of STPA-Priv is straightforward. There are a lot of steps which build up on each other, but they follow a simple logic: Define privacy parameters (like adverse consequences), find vulnerabilities, negate vulnerabilities to create constraints and then find control actions that violate these constraints. During the application we often recognized that we might have to add some things to previous steps and then step through all following steps again. This was no problem so STPA-Priv can and should also be applied iteratively if knowledge about the system is gained during or after the application. Despite this simple logic we found the application not easy. This is because to understand what we have to do in each step, a lot of additional information about the process has to be gathered. This information, for example how to choose a threat catalog or how to create the control structure model, is currently not included in the available literature. The most convenient place would be if the XSTAMPP tool would offer this information to the user e. g. in form of a tutorial for each step. Anyway a formal documentation of the method would be a good start. Currently XSTAMPP offers for the steps that require listing artifacts (like adverse consequences or control actions) only very humble tabular input forms. Nevertheless, a nice feature of XSTAMPP is the visual editor for the control structure model. We know that for the safety analysis this control structure representation can already been used to formally validate the model against the safety constraints. But this is currently not tailored for the privacy analysis. Also, it misses vital information on how the control structure model components have to be used in a privacy analysis.

Next steps: We want to compare STPA-Priv to the LINDDUN privacy threat modeling to find out if one method finds privacy risk that the other method does not and/or if one method requires less effort and resources. Also, we want to create formal documentation for STPA-Priv and make XSTAMPP more self-contained by adding its own privacy threat catalog.

Acknowledgment: This work was financed by the Baden-Württemberg Stiftung.

References

[1] P. Middleton, P. Kjeldsen, and J. Tully, "Forecast: The Internet of Things, Worldwide, 2013," Gartner Research, Technical Report G00259115, 2013.

[2] S. S. Shapiro, "Privacy Risk Analysis Based on System Control Structures: Adapting System-Theoretic Process Analysis for Privacy Engineering," in *Proceedings of the 2016 IEEE Security and Privacy Workshops*, ser. SPW '16, IEEE, 2016, pp. 17–24.

[3] K. Lu *et al.*, "Checking More and Alerting Less: Detecting Privacy Leakages via Enhanced Data-flow Analysis and Peer Voting," in *Proceedings of the 22nd Annual Network and Distributed System Security Symposium*, ser. NDSS '15, Internet Society, 2015, 4:1–4:15.

[4] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "PiOS : Detecting privacy leaks in iOS applications," in *Proceedings of the 18th Annual Network and Distributed System Security Symposium*, ser. NDSS '11, Internet Society, 2011, pp. 1–15.

[5] W. Enck *et al.*, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, 5:1–5:29, 2014.

[6] K. Wuyts, R. Scandariato, and W. Joosen, "LIND(D)UN privacy threat tree catalog," KU Leuven, Department of Computer Science, Technical Report CW 675, 2014.

[7] ——, "Empirical evaluation of a privacy-focused threat modeling methodology," *Journal of Systems and Software*, vol. 96, pp. 122–238, 2014.

[8] K. Wuyts and W. Joosen, "LINDDUN privacy threat modeling: A tutorial," Department of Computer Science, KU Leuven, Tech. Rep., 2015.

[9] N. G. Leveson, *Engineering a Safer World Systems Thinking Applied to Safety*. Cambridge, MA, USA and London, England: MIT Press, 2011.

[10] W. Young and N. G. Leveson, "An Integrated Approach to Safety and Security Based on Systems Theory," *Communications of the ACM*, vol. 57, no. 2, pp. 31–35, 2014.

[11] I. J. Nagrath and M. Gopal, *Control Systems: Engineering*. Kent, London: Anshan Publishers, 2008.

[12] A. Hazeyama, H. Washizaki, N. Yoshioka, H. Kaiya, and T. Okubo, "Literature survey on technologies for developing privacy-aware software," in *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, 2016, pp. 86–91.

[13] A. Abdulkhaleq and S. Wagner, "XSTAMPP 2.0: New improvements to XSTAMPP including CAST accident analysis and an extended approach to STPA," in *Proceedings of the Fifth MIT STAMP/STPA Workshop*, ser. STAMP Workshop '16, MIT, 2016, pp. 1–5.

[14] D. Siewiorek, "Generation Smartphone," *IEEE Spectrum*, vol. 49, no. 9, pp. 54–58, 2012.

[15] M. Knöll, "Diabetes City: How Urban Game Design Strategies Can Help Diabetics," in *Electronic Healthcare: First International Conference, eHealth 2008, London, UK, September 8-9, 2008. Revised Selected Papers*, D. Weerasinghe, Ed. Berlin, Heidelberg: Springer, 2009, pp. 200–204.

[16] ——, ""On the Top of High Towers ..." Discussing Locations in a Mobile Health Game for Diabetics," in *Proceedings of the 2010 IADIS International Conference Game and Entertainment Technologies*, ser. MCCSIS '10, IADIS, 2010, pp. 61–68.

[17] M. Knöll, M. Moar, S. Boyd Davis, and M. Saunders, "Spontaneous Interventions for Health: How Digital Games May Supplement Urban Design Projects," in *Technologies of Inclusive Well-Being: Serious Games, Alternative Realities, and Play Therapy*, A. L. Brooks, S. Brahnam, and L. C. Jain, Eds. Berlin, Heidelberg: Springer, 2014, pp. 245–259.

[18] M. R. Calo, "The boundaries of privacy harm," *Indiana Law Journal*, vol. 86, no. 3, pp. 1131–1162, 2011.