Socio-technical co-Design for accountable autonomous software*

Ayan Banerjee, Imane Lamrani, Katina Michael, Diana Bowman, Sandeep K.S. Gupta
Arizona State University
{abanerj3,ilamrani, katina.michael, diana.bowman, sandeep.gupta}@asu.edu

Abstract

Recently, efforts to regulate software and make organizations and individuals more accountable for its consequences have increased. Traditionally, the human-in-the-loop (operator, user, or bystander) is usually blamed for undesirable behavior of software systems in the real world. This is due to the limitations of the user-centered design approach where an average user's mental model (MM) is adopted. The core belief in this paper is that user-centered design must incorporate a wider lens of stakeholder interactions using socio-technical ecosystems being inclusive of users from various backgrounds and consulting with certifiers, manufacturers, and regulatory agencies for a given jurisdiction. We envision a socio-technical co-design approach for the development of compliant autonomous socio-technical systems (ASTS), which can infuse novel interpretations of regulations based on the social, behavioral, and economic (SBE) background of users. We posit that an accountable software has three properties: 1) *operational transparency:* Amenable to monitoring relevant parameters for tacit knowledge (user's MM of the ASTS), 2) *operational adaptability:* The software can be configured to support evaluation of regulatory compliance with changing performance expectations and compliance perceptions, and 3) *operational interpretability:* The software can assist in generating feedback for guidance on the compliance properties of novel modes-of-operations – a consequence of dissonance between MM of the user and the system designer's view of users' MM.

1 Introduction

Compliance-by-design is useful to ascertain accountability in software design for autonomous socio-technical systems (ASTS) [Graafstra *et al.*, 2010]. The key feature of compliance is in the incorporation of regulations in software implementation. Recent initiatives such as the General Data Protection Regulation (GDPR) compliant-by-design ASTS are a testimony to this effort [Truong *et al.*, 2019; Winfield *et al.*, 2019]. A **user-centered design** approach is undertaken, where manufacturers focus on the users and their needs engaging the participant to understand their interpretation of regulation into a set of requirements for the operational characteristics of the ASTS software [Robertson *et al.*, 2019]. Specific software modules are then developed and tested to ensure compliance to the requirements.

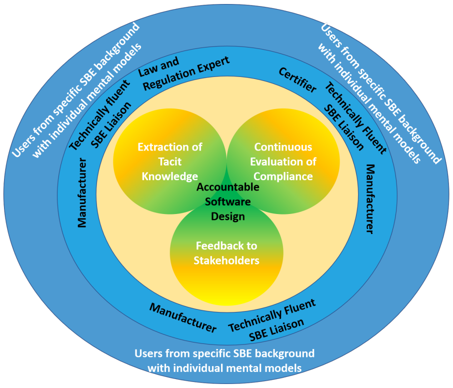
Users from specific SBE background with individual mental models
Law and Regulation Expert
Certifier
Users from specific SBE background with individual mental models
Technically fluent SBE Liaison
Technically Fluent SBE Liaison
Manufacturer
Extraction of Tacit Knowledge
Continuous Evaluation of Compliance
Accountable Software Design
Manufacturer
Feedback to Stakeholders
Manufacturer
Technically Fluent SBE Liaison
Users from specific SBE background with individual mental models

Figure 1: Socio-Technical ecosystem to address accountability in user-centered software design for autonomous systems.

The interpretation of regulation is a function of the social, behavioral, and economical background of a user and can differ significantly across population. However, the need for the development, certification and marketing of a minimum viable product (MVP) within time constraints often result in an interpretation of regulation that restricts compliance to limited usage configurations in the ASTS software. As such compliance may not hold when the social, cultural, behavioral and economic (SBE) background of a user results in novel interpretations of the regulation and non-certified usage configurations. In such cases, there currently exits no clear pathway towards ascertaining accountability to regulation. The core belief in this endeavor is that user-centered design must incorporate a wider lens of stakeholder interactions using socio-technical ecosystems being inclusive of users from various backgrounds and consulting with certifiers, manufacturers, and government agencies.

We envision that operational safety can be assured with a socio-technical co-design approach for the development of ASTS software that embeds regulatory compliance from the outset. In this approach, the user is considered as an *expert in the experience domain* and actively contributes to the accountability of the software design based on regulations, laws, applicable standards and risk management. The experi-

*This project is partly funded by DARPA AMP project.

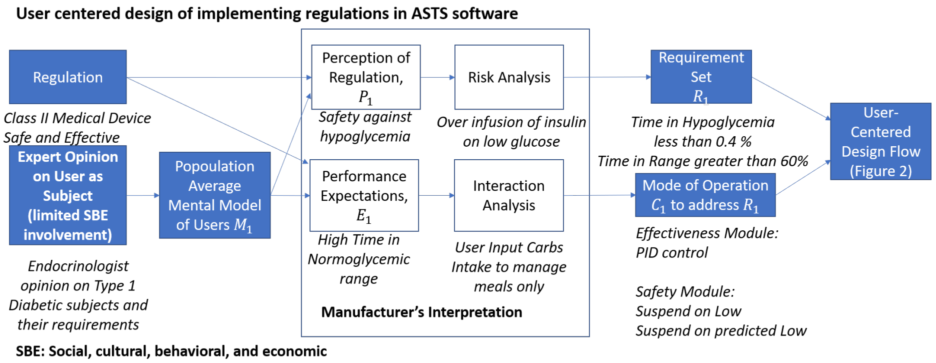
User centered design of implementing regulations in ASTS software
Regulation
Class II Medical Device
Safe and Effective
Expert Opinion on User as Subject (limited SBE involvement)
Endocrinologist opinion on Type 1 Diabetic subjects and their requirements
Popoulation Average Mental Model of Users M_1
Perception of Regulation, P_1
Safety against hypoglycemia
Risk Analysis
Over infusion of insulin on low glucose
Performance Expectations, E_1
High Time in Normoglycemic range
Interaction Analysis
User Input Carbs Intake to manage meals only
Manufacturer's Interpretation
Requirement Set R_1
Time in Hypoglycemia less than 0.4 %
Time in Range greater than 60%
Mode of Operation C_1 to address R_1
Effectiveness Module: PID control
Safety Module: Suspend on Low Suspend on predicted Low
User-Centered Design Flow (Figure 2)
SBE: Social, cultural, behavioral, and economic

Figure 2: State-of-Art user-centered approach towards accountable software design, Limited SBE involvement.

ence of a user is a function of the exposures [Robertson *et al.*, 2017] and the user's mental model of the ASTS. The mental model is directly affected by social (e.g., age, gender, education levels), cultural (e.g. customs, ethnicity, spritual beliefs, and practices), behavioral (e.g. perceived risks and benefits of adoption of ASTS) and economic background (e.g. affordability and insurance coverage) and is tacit knowledge embedded in the operational characteristics of a deployed ASTS software [Robertson *et al.*, 2017].

The **key feature** of the socio-technical ecosystem (Figure 1) is the introduction of a trans-disciplinary liaison as a stakeholder representing the user population in the user-centered design approach. A liaison acts as intermediary interlocutor between the user population and the other stakeholders who can: a) provide tacit knowledge to the other stakeholders about the effects of social, cultural, behavioral and economic characteristics of user cohorts on interpretation of perception of compliance and performance expectations from an ASTS, and b) explain compliance and operational characteristics of the ASTS software to the user to ethically align their interactions with the ASTS software [Michael *et al.*, 2021]. Socio-technical co-design attempts to improve software accountability with respect to regulation using a three pronged approach (Figure 1):

a) **Extraction of tacit knowledge:** A trans-disciplinary team (e.g. computer scientists, CSE and SBE scientists) attempts to extract conceptualization of user's mental model and novel usage configurations of the software from continuously monitored operational characteristics of the software.

b) **Certification game:** Continuous evaluation of novel usage configurations for regulatory compliance through a certification game between stake holders.

c) **Feedback to stake holders:** Generating feedback for the actors and stake holders to reshape mental models and improve perception of accountability.

2 Accountability issues in state-of-art

In the state-of-art user-centered approach (Figure 2) the manufacturer first develops a MVP to meet a technological and market demand gap. The required regulations are then carefully evaluated by the manufacturer in collaboration with a law expert, taking into account the regulatory agency guidance. For example, semi autonomous medical devices such as closed loop blood glucose control systems (Minimed 670G by Medtronic) are classified as class II devices by Food and Drug Administration (FD& C Act) and are accountable to 510K pre-market approval. The regulatory law states that the device should not raise different questions of safety and effectiveness than another legally marketed device.

In the user-centered approach (Figure 2) the manufacturer utilizes expert advice from secondary sources of evidence to make a population average mental model that translates the regulatory law into requirements on the operational characteristics of the software. For the example case, the 510K pre market approval regulation is converted to two requirements: a) **effectiveness**, percentage of time in normo-glycemic range (TiR) greater than that reported in sensor augmented pump (SAP), and b) **safety**, low percentage of time in hypoglycemia than in SAP [Berget *et al.*, 2020].

Through interaction and risk analysis the manufacturer then builds software add-ons to the MVP to address regulations. For the case of Minimed pump, this is a supervisory control software component to automatically deliver insulin between manually announced meals to address effectiveness, and two safety modules, suspend on low and suspend on predicted low. The manufacturer then collaborates with domain experts to conduct in-the-field studies to collect data on the compliance and performance properties. The data driven compliance argument is then submitted to the certifier (FDA in case of Minimed 670G) for regulatory approval. The state-of-art user-centered design approach has the following drawbacks:

Limited reconciliation pathways can result in unmet expectations: Population average models may not be applicable for a given user. In such scenarios, the performance expectations may not be met. In the current state-of-art the pathway towards a reconciliation is vague and unlikely.

For example, studies [Berget *et al.*, 2020] show that many Medtronic 670G users (actors) experience significantly less TiRs than that observed in clinical studies used for showing regulatory compliance [Berget *et al.*, 2020]. A primary reason for this is that several actors spend less than 60% of time per day in auto mode, where the PID controller is active. The auto mode exits are triggered by two sources: a) the safety modules suspend insulin delivery when glucose levels reach or are predicted to reach low values, and b) the auto mode defaults to safe basal mode nearly 15% of the time when glucose levels are high for a long duration. The safe basal mode injects a constant level of basal insulin and does not utilize the PID algorithm. Such usage artifacts were not observed in clinical studies which report 95% time in auto mode [Berget *et al.*, 2020]. This is a violation of the effectiveness component of regulation and currently there is no pathway towards a reconciliation. In fact, this pushes the user towards initiatives such as Do-It-Yourself (DIY) [Ahmed *et al.*, 2020] insulin pumps that allow unsatisfied users to design their own control software.

Unmet expectation may lead to ethically misaligned user behavior: Ethical values are a function of social and behavioral background of an user. Unmet expectations from a software can often trigger the user to subvert compliant operation of the software. For example, a user can have a performance expectation to minimize post prandial glucose level instead of increasing TiR. (Post prandial hyperglyemia has a strong positive correlation with HbA1C levels [Ferenci *et al.*, 2015], the gold standard metric for Type 1 Diabetes.) In such a scenario, the safety module of suspend on low or predicted low may induce unnecessary auto mode exits resulting in lesser insulin delivery and higher post prandial hyperglycemia. High post prandial glucose levels is often managed through *phantom carbs* [Weaver and Hirsch, 2018], where the user announces a meal to the device, without actually consuming it. The purpose is to trick the device to administer a heavy bolus insulin.

An unpredictable unethical user behavior results in unresolved accountability: In user-centered design approach, the definition of ethical behavior is often unclear. This is because the user-centered design does not consider value sensitive aspects of an user, which are functions of the social behavioral and economic background. As such the user may often be unaware of ethically misaligned interaction. On the other hand, ASTS software has been only certified for the specific use case. Since the interaction scenarios between the user and the ASTS are potentially limitless, there may not be specific guidance for a given ethically misaligned interaction. For example, the case of phantom carbs is never mentioned in Minimed 670G user manuals or safety instructions. Since, the device is unaware of the status of the meal consumption, such behavior can lead to severe hypoglycemia and potentially death.

In an user centered design approach, when compliance fails due to an unexpected wrongful use case, accountability becomes hard to resolve.

3 Socio-technical accountable software design

Social counter part consists of citizens as users, operators, oversight, certifiers and manufacturers. Whereas the technical counter part consists of the software system. It is a co-design approach that involves iterative knowledge sharing among two models: a) mental model that encodes the social aspects of the design and b) operational model that encodes the technical aspect of the design.

Operational Model: An autonomous system operational characteristics is captured by the function $Y(t) = S(X(t), t)$, where $Y(t)$ is the response of the autonomous system and $X(t)$ is the external input to the system. A susbset of $X(t) \bigcup Y(t)$ is monitored in the real time using sensors deployed with the system.

Manifestations of operational model: The software of an autonomous system can be abstracted using various formal structures such as finite state machine or hybrid systems. Such models can express the input output relationship in an autonomous software by combining discrete modes and dynamic variations of system parameters over time.

Example: In case of the Medtronic 670G semi-autonomous insulin control system, an operational model can be a hybrid system. The finite state machine part of the hybrid system models the discrete modes such as basal, auto, correction bolus, and meal bolus modes of the software, whereas each state can express the control decisions as a set of differential equations. The transitions between each mode is governed by external events such as meal intake or internal events when blood glucose levels are within specific ranges.

Mental Model: It is the perceived operational characteristics and performance expectations of a citizen user. The mental model guides the interaction of the user with the exposures of an autonomous system to achieve the expected performance for a given environmental context. It also guides the interpretation of a regulation into requirements on the operational model. Hence, it is a connector between the user, certifier, oversight, and manufacturer.

The mental model is influenced by the socio-cultural background of a citizen. It is denoted by the notation $M(S(.))$ and can be vague, imprecise, and can dynamically change.

Conceptualization of mental model of a citizen: Mental model is in the user's mind, but is embedded in the actions of the user for a given response from the controller. Hence, a conceptualization of the mental model can be derived by observing the action (external input $X(t)$) and response (output $Y(t)$) in a deployed autonomous system.

The first step towards building a conceptualization of a mental model is exposure analysis. It determines the subset $E \subset X$ of exposures.

The second step is to derive a precise mathematical function, $E(t) = C(M(S(.)), S(X(t), t), Y(t))$ that expresses the temporal sequence of stressors applied by the user on the exposures. It characterizes the composite effect of the mental model, operational model, and the observed response from the controller on the inputs from the user.

Manifestations of mental model conceptualizations: In this research, we will consider two types of manifestations: a) surrogate models, finite state machine, temporal logic or hybrid system based expressions of $C(...)$ and b) task action mapping models, that expresses the sequence of stressors from the user as a language.

Example: Meal intake is manually managed in the Medtronic 670G system. Typically a routine meal intake is expected and the insulin delivery in the controller is dependent on this routine. However, cultural/religious practices such as ramadan can affect meal routines. Such changes in routines can be expressed using a temporal logic model, where the finite states are events and the temporal properties express the change in meal timings for a given event.

The mental model that causes announcement of phantom meal can be conceptualized using a state machine based surrogate model that is similar to the operational model of the Medtronic 670G but has an extra phantom meal mode to express fake carb entry.

Regulatory statute: It is a textual description of a property that the autonomous system should abide by. It is often a general guidance and is intended to apply for a broad set of citizens with varied mental models. Compliance with regulation can be qualified by limiting the context of usage that includes external environment and responses from the citizen.

Regulatory Requirements: A regulatory statute can be interpreted as a envelop (max, min) on the observed output variables ($Y(t)$) of an autonomous system for a specific use case taking into account the mental model of the citizen, and external environmental context of usage. The autonomous system is expected to meet the requirements in order to comply with the regulatory statute.

Accountability in software: Accountable software system is a system that is flexible to change and context, expressibe enough to capture policy, *provably and certifiably com-*

pliant, and transparent/auditable to policymakers and stakeholders while maintaining the manufacturer's competitve advantage by not revealing their trade secrets. In this proposal, we define accountable software systems as systems that enable stakeholders to ensure and prove compliance in the field of operation. Three properties constitute accountability:

a) *operational transparency:* The system software should provision for: a) monitoring of input $X(t)$ and response $Y(t)$ parameters, and b) an explanation interface that can extract relevant knowledge from the observed parameters for different stakeholders in the software development, deployment, certification, and regulation process.

b) *operational adaptability:* When a change in interpretation of the regulatory statute results in new context sensitive requirements, the software provides mechanisms to monitor relevant variables necessary to evaluate compliance with the new requirements.

c) *operational interpretability:* In the event of compliance failure, the cause can be attributed to a specific software component or citizen interaction, or environmental conditions.

4 Pillars of Socio-Technical Co-Design

Continuous monitoring is key to the socio-technical co-design approach. The input output set $(X(t), Y(t))$ is monitored for a deployed software which is regulatory compliant for the population average mental model. The following pillars should then guide our socio technical co-design approach.

4.1 Pillar 1: Extraction of Tacit Knowledge

Mental models are tacit knowledge for ASTS software and are layered as depicted by the Iceberg systems thinking model (Figure 3) [Webb *et al.*, 2008]. The lowest level (L4) is the farthest from observable aspects of the software and represents the actor's assumptions about the software system before interacting with it. Such assumptions is built based on information acquired from reports, manuals, policies, regulations, and media. This is also affected by the actor's social, economic, and cultural background.

L3 represents the refined mental model of the actor after its interaction with the software. At this level, actors adapt (contradict or enhance) beliefs they have about the system based on their own experience with the system. The result of such an interaction can be interpreted differently from one actor to the other depending on the theory they use to create meanings.

The next level L2 represents the trends and patterns of the holistic system's operational characteristics arising due to the interaction of the actor. Information on this layer is typically available to the manufacturer and is used to evaluate performance and regulatory compliance.

Finally, L1 represents observed events that are caused by the holistic system's operational characteristics.

Value Sensitive Interpretation of Regulatory Law

This is tacit knowledge extraction from Level L4. There are no observable parameters that can be utilized to derive or validate the mental model of the actor. However, there are a rich body work in the social sciences domain that have already studied various social systems.

Potential Methodology: Social values guide an actor's performance expectation as well as perception of compliance to regulatory law. User engagement should involve interviews to identify two aspects:

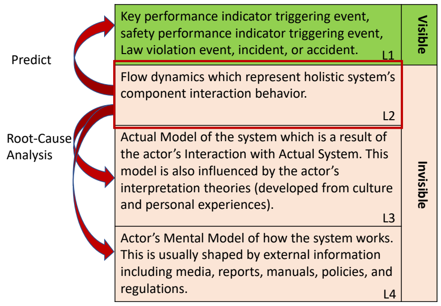
Predict
Root-Cause Analysis
Key performance indicator triggering event, safety performance indicator triggering event, Law violation event, incident, or accident.
L1
Visible
Flow dynamics which represent holistic system's component interaction behavior.
L2
Actual Model of the system which is a result of the actor's Interaction with Actual System. This model is also influenced by the actor's interpretation theories (developed from culture and personal experiences).
L3
Actor's Mental Model of how the system works. This is usually shaped by external information including media, reports, manuals, policies, and regulations.
L4
Invisible

Figure 3: Mental model levels.

- **Basic interpretation of Law:** This should focus on what a given regulatory statute actually mean for a given actor. This is irrespective of any ASTS solution in the domain.
- **Interpretation for a given system:** In this task, a thorough description of the ASTS should be provided to the actor through video demonstrations and advertisements. The actor should then be interviewed about their performance expectation from the system and their perception of compliance for the given system.

To interpret the perception of compliance from the interview data a collaborative effort between the SBE and technical experts can be undertaken utilizing the **System Theoretic Process Analysis (STPA)** formalism [Leveson *et al.*, 2003]. STPA is a hazard analysis technique that provides guidance to engineers in the design process and is widely used by industries including autonomous vehicles, Advanced Driver Assistance Systems (ADAS), unmanned aerial vehicle, nuclear power plants and many other safety-critical software systems [Leveson *et al.*, 2003]. STPA accounts for a broad range of causal factors including dysfunctional system interactions, incomplete/incorrect actors' mental models, and flawed design. STPA has been extended to analyze causal factors of undesirable events arising from flawed actors' mental models. The actor's mental model encompasses the mental model of the environment (legal, social, and economical contexts) where the system operates, mental model of the system which is built by the actor using information from reports, media, and any educative documents, and the mental model of the system's expected behavior which describes the actor's expectations of how the system will behave. The actor adapts the expected behavior model based on sensory inputs and feedback from real world interaction. STPA also analyzes how humans may adapt their mental models from interaction with the system and include the repercussions of flawed model adaptations. As shown in Figure 4, the actions and behavior of the human-in-the-loop is influenced by the human's interpretation of a variety of external inputs and how the human assimilates the input information into mental model representations.

Conceptualization of Mental Models

Information from Level L3 can be utilized to conceptualize mental models. The L3 information is in the form of external inputs to the ASTS obtained from the actors in a context rich environment. Understanding the learning model of the actor is a significant step towards conceptualization of mental model. Literature in human computer interface (HCI) research suggest that there are eight different types of human learning models [Gentner and Stevens, 2014] -

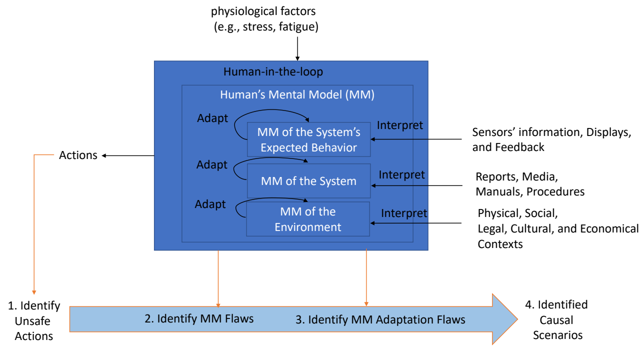
physiological factors
(e.g., stress, fatigue)
Human-in-the-loop
Human's Mental Model (MM)
Adapt
MM of the System's Expected Behavior
Interpret
Sensors' information, Displays, and Feedback
Actions
Adapt
MM of the System
Interpret
Reports, Media, Manuals, Procedures
Adapt
MM of the Environment
Interpret
Physical, Social, Legal, Cultural, and Economical Contexts
1. Identify Unsafe Actions
2. Identify MM Flaws
3. Identify MM Adaptation Flaws
4. Identified Causal Scenarios

Figure 4: STPA Extension to Include Human Factors.

a) Strong Analogy, where the actor finds a strong similarity to another software that they have prior experience with.
b) Surrogate models, where the actor derives notational analogue such as a finite state machine model of the mechanism of the software.
c) Mapping models, where the actor makes a table of actions and responses.
d) Coherence models, where the actor makes a logical schema of operational characteristics of the software which helps the actor to remember how to interact. This model is vague because if the response of the software is not coherent with the schema, the software feature maybe forgotten.
e) Vocabulary model, where the actor creates a grammar that expresses the temporal sequence of actions that are required to be performed by the actor to elicit a given response.
f) psychological grammar, where the actor finds an analogy with the grammar of their native language.
g) Problem space, where the responses of the actor is modeled as a solution to challenge question from the software.
h) Commonality model, where actor actions are considered as processes sharing the same data structure.
Potential Methodology: Data collected from each interview conducted should be utilized to instantiate the conceptualizations with appropriate models. The models have to be unambiguous and complete. State reachability analysis can be utilized to check for undefined reachable states.

Decoupling mental model and operational model

The operational characteristics of an ASTS is a composite result of a closed loop execution of two processes: a) mental model guided interaction of the actor with the software, and b) response of the software to the actor's input. The operation is almost always initiated by the actor, which is sensed using sensors, the software then reacts to the input by generating a response, that is provided as feedback to the actor, closing the loop. As such this two way causal relationship can be extracted by analyzing the dependencies of the input set $X(t)$ and response set $Y(t)$.
Potential Methodology: The problem is an evolved form of system identification where the observable variables are controlled by two parallel co-operating processes. For the Medtronic 670G example, the meal management software is initiated by the actor by providing a carbohydrate value as an input. The bolus wizard software component then computes the amount of insulin to infuse and administers a bolus insulin. Subsequently, based on the continuous glucose monitor (CGM) sensor values it executes a PID control strategy to continually infuse micro bolus insulin to control the blood glucose. For this operational context the input set $X(t)$ causes the output response $Y(t)$.

On the other hand, if the meal management strategy leads to prolonged hyperglycemia, then the Minimed 670G triggers the correction bolus mode. In this mode, the actor is prompted to take a Glucosemeter reading and enter it to the system. Here the response set $Y(t)$ causes the input set $X(t)$.

An essential step towards decoupling the mental model with the operational model is to derive the causal relationships between $X(t)$ and $Y(t)$. These causal relationships can be obtained from an algorithmic description of the ASTS.

4.2 Pillar 2: Certification game

The purpose of this pillar is to evaluate compliance for new modes of operation or under the new value sensitive requirements. However, the inherent assumption is that all required data will be monitored and shared between stakeholders. This is often not feasible, because of the sensitive nature of data and potential violation of regulatory laws such as HIPAA, GDPR, and patent laws. The operational adaptability component of accountable software necessitates that the manufacturer participate in a certification game with the certifier. The objective of the game for both manufacturers and certifiers is to evaluate the compliance of either: a) the new mode of operation, or b) the present mode of operation under novel value sensitive requirements. The game consists of a sequential execution of steps initiated by manufacturer and continued by the certifier.
Step 1: Manufacturer Share Level L1 information. This information sharing is a part of continuous compliance check property of accountable software.
Step 2: Certifier checks learnability: The certifier then attempts to mine tacit knowledge using theories in Thrust 1. The result of this step is a guidance to the manufacturer regarding potential sharing of Level 2 and Level 3 information.
Step 3: Certifier evaluates compliance properties: With the information currently available to the certifier, it uses formal software analysis tools and techniques to extrapolate compliance properties to the new mode of operation, or to the present mode of operation with value sensitive requirements.
Step 4: Manufacturer Share Level L3 or L2 information. The manufacturer takes the learnability and compliance guidance and makes a decision to lawfully share L3 or L2 information to the manufacturer. In the process the manufacturer performs a cost risk and benefit analysis and may chose to share a different set of information than that required by the certifier or nothing at all.
Step 5: Repetition of Step 2 by Certifier. If new information is shared the certifier repeats Step 2. If no information is shared, the compliance extrapolation obtained in Step 3 is issued as guidance to the actors.

Evaluating Learnability of Tacit Knowledge

An ASTS design consists of an actor model that embeds the assumed interactive behavior of the actor with the remaining components of the system, an environment model usually represented by a set of ordinary differential equations (ODEs) governing the high-dimensional system, and the controller model that utilizes participants and environment models along with sensor data to determine control actions to satisfy a predefined goal function. We consider the fact that contingencies in participant behavior and novel/unseen Environment-Controller-Actor interaction scenarios can be detected as a deviation from the expected evolution of the system's dynamics. An ultimate solution to enhance safety is to learn changes in the variation of the physical dynamics within each controller mode and verify whether theses changes represent a potential hazard to the system using state-of-the-art safety verification techniques that are employed during the system's engineering[Henzinger *et al.*, 1997].

Potential Methodology: An intuitive approach to solve this problem is to directly use residual neural networks or ODE nets to learn a generative latent model of the dynamical system. Although deep learning techniques can learn model parameters but they require large amount of data which may not be available in practical deployment scenarios. The complexity and resources required for learning such models is proportional to the number of function evaluations performed in the forward pass, i.e. the size of the governing ODEs, number of unknown parameters, and number of latent variables. Such large I/O data may not be available due to several reasons including data logging insufficient capacity or a high learning frequency required for the safety evaluation of the semi autonomous system. Another approach is to utilize contextual conditions to reduce the model learning to a set of linear or polynomial regression analysis. Contextual information of data can provide initial and asymptotic conditions of every operational mode to simplify the operational model learning.

4.3 Pillar 3: Reshaping mental model

The third arm of accountability in software is interpretability of compliance properties to the stake holders. Given the diverse goals and backgrounds of the stake holders, effective feedback should be a function of the objectives of each stake holder and should not be specific to an instance of the operation of the ASTS software. Concept level feedback to stake holder is essential for better communication. In the process of forming a mental model of any system or process, the human learns general concepts that is applicable to any instance of the system or process. According to human learning theories, a feedback in terms of concepts used is effective in creating memories and taking actions to complete objectives.

Feedback to Manufacturer

The certifier in the certification game provides feedback to the manufacturer in terms of the operational model and its compliance properties. However, such feedback may not directly enable the manufacturer to identify the components that can be monitored or adapted to facilitate compliance evaluation and satisfaction. The manufacturer is familiar with the software design and typically develops architectural models of the software before implementation. Hence if a feedback from the certifier is in terms of the these architectural model components then it is one step closer to identifying the next steps towards accountability.

Value Sensitive feedback to actor

Contrary to the manufacturer, feedback to the actor may not be concretely expressed in terms of some objective components such as software code. While feedback to the manufacturer is uniform, for an actor the feedback should be diverse commensurate with their social, behavioral, economic and cultural background.

5 Conclusions

Autonomous systems are failing to maintain and ensure compliance to regulations when deployed in practice resulting in loss of public trust. The Boeing 737 Max 8 has been grounded worldwide, increasing incidence of crashes involving autonomous cars resulting in lawsuits against companies, and average Type 1 diabetic subjects having a decreasing amount of time that they are spending in closed loop auto mode. Research in socio-technical co-design of ASTS software will result in an improved performance of autonomous systems in practical deployments and providing higher compliance guarrantees to stake holders. This approach relies on collaboration between CSE and SBE scientists with the aim of analyzing the bi-directional impact between the software system's operation and the legal, social, and behavioral contexts of the system's operating environment.

References

[Ahmed *et al.*, 2020] S. H. Ahmed, D. L. Ewins, J. Bridges, A. Timmis, N. Payne, C. Mooney, and C MacGregor. Do-it-yourself (diy) artificial pancreas systems for type 1 diabetes: Perspectives of two adult users, parent of a user and healthcare professionals. *Advances in therapy*, 37(9):3929–3941, 2020.

[Berget *et al.*, 2020] Cari Berget, Laurel H Messer, Tim Vigers, Brigitte I Frohnert, Laura Pyle, R Paul Wadwa, Kimberly A Driscoll, and Gregory P Forlenza. Six months of hybrid closed loop in the real-world: An evaluation of children and young adults using the 670g system. *Pediatric diabetes*, 21(2):310–318, 2020.

[Ferenci *et al.*, 2015] Tamás Ferenci, Anna Körner, and Levente Kovács. The interrelationship of hba1c and real-time continuous glucose monitoring in children with type 1 diabetes. *Diabetes research and clinical practice*, 108(1):38–44, 2015.

[Gentner and Stevens, 2014] Dedre Gentner and Albert L Stevens. *Mental models*. Psychology Press, 2014.

[Graafstra *et al.*, 2010] A. Graafstra, K. Michael, and M. G. Michael. Social-technical issues facing the humancentric rfid implantee sub-culture through the eyes of amal graafstra. In *2010 IEEE International Symposium on Technology and Society*, pages 498–516, 2010.

[Henzinger *et al.*, 1997] Thomas A Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. Hytech: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer*, 1(1-2):110–122, 1997.

[Leveson *et al.*, 2003] Nancy G Leveson, Mirna Daouk, Nicolas Dulac, and Karen Marais. Applying stamp in accident analysis. 2003.

[Michael *et al.*, 2021] K. Michael, R. Abbas, R. A. Calvo, G. Roussos, E. Scornavacca, and S. F. Wamba. Smart infrastructure and technology systems ethics. *IEEE Transactions on Technology and Society*, 2(1):2–3, 2021.

[Robertson *et al.*, 2017] L. Robertson, A. M. Aneiros, and K. Michael. A theory of exposure: Measuring technology system end user vulnerabilities. In *2017 IEEE International Symposium on Technology and Society (ISTAS)*, pages 1–10, 2017.

[Robertson *et al.*, 2019] Lindsay J Robertson, Roba Abbas, Gursel Alici, Albert Munoz, and Katina Michael. Engineering-based design methodology for embedding ethics in autonomous robots. *Proceedings of the IEEE*, 107(3):582–599, 2019.

[Truong *et al.*, 2019] Nguyen Binh Truong, Kai Sun, Gyu Myoung Lee, and Yike Guo. Gdpr-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, 15:1746–1761, 2019.

[Weaver and Hirsch, 2018] Kathryn W Weaver and Irl B Hirsch. The hybrid closed-loop system: evolution and practical applications. *Diabetes technology & therapeutics*, 20(S2):S2–16, 2018.

[Webb *et al.*, 2008] David C Webb, Nina Boswinkel, and Truus Dekker. Beneath the tip of the iceberg: Using representations to support student understanding. *Mathematics teaching in the middle school*, 14(2):110–113, 2008.

[Winfield *et al.*, 2019] A. F. Winfield, K. Michael, J. Pitt, and V. Evers. Machine ethics: The design and governance of ethical ai and autonomous systems [scanning the issue]. *Proceedings of the IEEE*, 107(3):509–517, 2019.