sciendo

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 63 (76) 2020

DOI: 10.2478/slgr-2020-0032

Giuseppe Primiero
University of Milan
Department of Philosophy
e-mail: giuseppe.primiero@unimi.it
ORCID: 0000-0003-3264-7100

Balbir Barn
Middlesex University London
Department of Computer Science
e-mail: b.barn@mdx.ac.uk
ORCID: 0000-0002-7251-5033

Ravinder Barn
Royal Holloway, University of London
Department of Law and Criminology
e-mail: r.barn@rhul.ac.uk
ORCID: 0000-0002-4489-2632

VALUE-SENSITIVE CO-DESIGN FOR RESILIENT INFORMATION SYSTEMS

Abstract. In Information Systems development, resilience has often been treated as a non-functional requirement and little or no work is aimed at building resilience in end-users through systems development. The question of how values and resilience (for the end-user) can be incorporated into the design of systems is an on-going research activity in user-centered design. In this paper we evaluate the relation of values and resilience within the context of an ongoing software development project and contribute a formal model of co-design based on a significant extension of Abstract Design Theory. The formal analysis provides a full and clear-cut definition of the co-design space, its objectives and processes. On the basis of both, we provide an abstract definition of resilient system (for the end-user). We conclude that value-sensitive co-design enforces better resilience in end-users.

Keywords: Abstract Design Theory, Value Sensitive Design, Co-Design, Resilience.

1. Introduction

The current landscape of the Philosophy of Computer Science has progressively moved from a view of computational systems has having a dual ontology relying on the distinction between hardware and software

(Moor, 1978; Colburn, 1999; Turner, 2011), to one where a layered ontology is at stake, which includes intentions, specifications, algorithms, programming and machine code languages, and finally executed computational processes (Primiero, 2016, 2020). Under this latter view, key moral and epistemic values such as privacy, security and autonomy acquire full right of existence in the ontological and epistemological analysis of software systems. Failure to address these demands may have a detrimental effect on the resilience of end-users, namely their ability and willingness to accept and keep using a new technology. While much progress in value-sensitive co-design has been made, the nuances of how values of security, privacy and autonomy become incorporated into the design process remains an ongoing challenge. In general, the effects of value-based design on theory and methodology is yet largely unexplored, including its relevance for explanation in software development.

Resilience is one of such notions. A working definition from the Organisation for Economic Co-operation and Development (OECD) that should suffice for the present purposes is the following:

> [Resilience is] the ability to cope with changes in capacity, effectiveness or legitimacy. These changes can be driven by shocks... or through long-term erosions (or increases) in capacity, effectiveness or legitimacy. (OECD, 2008).

In Information Systems (IS) research and Software Engineering (SE), resilience has often been viewed through the lens of non-functional requirements and examined in terms of scalability, reliability, maintainability and availability. What has not been easily understood or investigated in the context of IS research, is how resilience is manifested in end-users as a result of an intervention such as the introduction of a system. The many variants of IS evaluation models such as the Technology Acceptance Model (Davis, 1993) and Unified Theory of User Acceptance of Technology (Venkatesh et al, 2003) do not include resilience as a determining factor and therefore resilience is not assumed to contribute to the acceptance of systems. Moreover, the fact that no proper formal counterpart to these models and theories has been provided, makes it even harder to establish resilience as a property of design methods.

A further concept is the notion of *value* (beyond a monetary sense), which at least has a history of research investigating the implications of value sensitive concerns to the design of systems. We suggest that value is intrinsically related to resilience by proposing that addressing value based concerns in IS systems design can encourage and engender resilience in the end-users of systems. The question of how values and resilience (for the end-user) can be incorporated into the design of systems is an on-going research activity in user centered design but remains largely absent from method engineering research and its formal area.

In this paper we first offer a critique of the intrinsic relationship between values and resilience by drawing upon extant literature (Section 2). We then evaluate that relationship within the context of an ongoing research project aimed at developing mobile apps for promoting better engagement between young people and their case workers in the UK youth justice system (Section 3). We finally propose a formal model of co-design based on a significant extension of Abstract Design Theory (Section 4) and argue how it justifies the claim that value-sensitive co-design enforces better resilience in end-users (Section 5).

2. Resilience and Values

In metallurgy and material sciences in general, resilience is a physical property of a material to bounce back to its normal shape after some deformation event (Callister, 1994). Ecologists have used resilience to refer to the ability of ecosystems to absorb and respond to disturbance (Holling, 1973). Psychologists use definitions that encompass two inter-dependent parts: adaptive functioning and exposure to risk or adversity (Wright et al, 2013). Moreover, resilience has been characterized as the positive capacity of individuals to cope with stress and catastrophic events and their level of resistance to future negative events (Erol at al, 2010). Resilience depends upon the existence of broadly three types of protective factors including: the characteristics of the individual (e.g. temperamental qualities, cognitive ability), the quality of the individual's relationships and broader ecological factors (Greenberg, 2006). The psychologist's view of resilience offers additional elaborations that resilience is not a general quality representing an individual's trait and that research needs to focus on the processes underlying individual differences in response to environmental hazards, rather than resilience as an abstract entity (Rutter, 2006). Interaction is key in such a process approach. Researchers have also noted how a person can demonstrate resilience in one domain but not another, or at one point in time but not another (Wright et al, 2013; Rutter, 2006). Thus resilience is both contextual and temporal. A consolidated view emerges that this lack of consistency or permanency of resilience over time or aspect of development, even within individuals, is evidence against conceptualizing resilience as a single quality or trait (Rutter, 2006; Wright et al, 2013; Ungar, 2013).

Opportunities for engendering resilience in the end user through the use of technology are becoming more widespread, particularly in the area of emergency and disaster management and there are numerous examples referring to the use of social media. For example, Twitter has been used to harness communities together for responding to disaster emergencies (Hughes and Palen, 2009; Shklovski et al, 2008; Gao et al, 2011). Of greater relevance is the study by Mark et al (2009) who note the inference of a relationship between resilience and values. They reported on how technology was adopted and used by citizens to be resilient during wartime. They identified properties of resilience: reconfiguring social networks, redundancy, proactive practices and repairing trust in information. Resilience seems thus to be also grounded on *values* recognition.

Early efforts at computerisation are characterised by an absence to address the notion of *value* beyond the narrow sense of the economic worth of an object and do not accommodate the dependencies on the interests and desires of human beings at large. Friedman et al (2006) identify certain values that are particularly pertinent to information systems development: ownership and property, privacy, freedom from bias, universal usability, trust, autonomy, informed consent, identity and others. *Values* have, to-date, been utilised in systems design largely through work in Participatory Design (PD, Bjerknes et al (1987)) and more latterly as “Co-Design” (Greenbaum and Kyng, 1991). Co-design involves potential (un-trained) end users working jointly with researchers and designers using tools provided to jointly create artefacts that lead directly to the end product (Sanders, 2000), becoming a dominant methodology in product and service design (Yoo et al, 2013). PD in its various forms has attempted to accommodate values within their methodological frameworks. For example, Contextual Design (Beyer and Holtzblatt, 1998) provides techniques for analysing cultural or political forces in the organisation that may impinge on roles to prevent or modify how work is done. Limitations on how values are managed within PD based approaches include issues that may arise as not all users can participate in decision making, or by conflicting values and preferences. In product development contexts, user involvement may be transitory and preferences and value considerations may not be evident in short discussions (Kujala and Väänänen-Vainio-Mattila, 2009).

Value-sensitive design (VSD) emerged to integrate moral values (and more broadly ethics) with the design of systems. A key premise of VSD is that it seeks to design technology that accounts for human values throughout the design process (over and beyond the identification of functionality and visual appearance) of systems. VSD has developed both methods and theory that incorporate particular values into technologies through conceptual, empirical, and technical investigations. A conceptual investigation of value-sensitive design involves questions about the stakeholders being affected by the design; the values being implicated; and to what extent the moral values have weight compared to non-moral values. An empirical investigation would take into account the human context and be used to measure the success of certain designs. A technical investigation focuses on how existing technological properties and mechanisms support or hinder human values. VSD recognizes two classes of stakeholders: direct and indirect. Direct stakeholders refer to parties – individuals or organisations – who interact directly with the computer system or its output. Indirect stakeholders refer to all other parties who are affected by the use of the system. VSD is also intended as an interactional theory, that is, values are neither embedded in the technology nor are they transmitted by external social forces. Instead, features that are designed may support certain values and hinder others. For example, the Microsoft Outlook calendar sharing feature supports an individual's accountability to an organisation but renders privacy difficult. VSD has further evolved to recognise specific value based inputs from users in a co-design space using a conceptual framework of designer prompts, stakeholder prompts co-operating and undergoing a process of reflection (Schön, 1983) to arrive at a shared design (Yoo et al, 2013). A key aspect of the approach is the incorporation of reflection into the design process and to provide a means for incorporating empirical data on values into the co-design space. In Yoo et al (2013), the traditional co-design core blends methods from value-sensitive design to structure the co-design engagement with inputs from stakeholders and considerations of values. Designer prompts entail materials that originate from expert designers and may comprise personas, scenarios or the use of envisioning cards. Stakeholder prompts originate from the end-users and may utilize value based scenarios addressing concerns such as unintended uses of the system; changes of the use of the system over time and so on. Values will be, typically, those derived from the list suggested in Friedman et al (2006). The reflection element of the model provides a way of representing how prompts may be generated by either a stakeholder or a designer as a result of joint participation in the co-design space.

We propose that there is an intrinsic link between accounting for values and engendering resilience in target end-user community. Values, we contend, can have a controlling effect on whether resilience will be enabled. By addressing value sensitive concerns in the design process, resilience as an outcome is more likely. Explorations of values and their linkage to design processes have, up to now, been relatively informal. We suggest that a formal model of value-sensitive co-design processes can help highlight such connection. We now evaluate this proposal in the context of an on-going research study.

3. Case Study: Young People in the UK Youth Justice System

The socio-technical context for this research concerns young people in the UK Youth Justice system. Research suggests that engagement with young offenders to help promote social inclusion and prevent re-offending remain key challenges for public policy and youth justice service providers (Smith et al, 2006). Yet currently, digital tools that could engender closer engagement and encourage co-creation between Youth Offending Team (YOT) case workers and young offenders are not available. There are no tools that utilise web 2.0 and mobile apps that specifically support young offenders in managing their personal situation and creating resilience within the youth justice system. Given that there are, annually, around 40,000 first time offenders and over 10,000 YOT workers in the UK, the need is significant (MoJ, 2013). Instead, the use of technology has largely focused on surveillance (Nellis, 2004) and supporting organisational structures and processes for data collection and information management. Neither of these uses is aimed at using technology for addressing the expressed needs and concerns of excluded groups and neither is focused on a positive and direct engagement with young offenders.

The MAYOT (Mobile Applications for Youth Offending Teams project, www.mayot.mdx.ac.uk) aims to produce a personalised mobile app for use by young people and their case workers in youth offending teams. The app is intended to provide relevant, timely information to a young person as well as features such as ease of access to their case history, relevant contacts such as professional networks, peer networks and family networks. Participants to the co-design activities for the app were drawn from three Youth Offending Teams (YOTs) in the UK, covering a mixture of inner-city, urban and rural areas. During the requirements process six co-design workshops were carried out with the first workshop serving as a pilot workshop. There were a total of 38 participants of which 24 were caseworkers and 14 were young people. Questionnaires were used as additional data collection mechanisms and a total of 65 youth justice professionals and young people contributed to that aspect. We also inspected case data of young people in the care of youth offending teams as well as strategy documentation. The scientific

Table 1

Features / Prompts

Feature Description	Originating Type	Resilience protective factor	Value	Status
Text based Reminders Mechanism to automate SMS based reminders to the young person and their close relatives.	Designer	Individual relationships/ Regulatory activities	None	Accepted
Activities such as photo-blogging; daily diaries.	Designer	Regulatory activities	Privacy (inappropriate use of photographs)	Likely Acceptance
Identified Goals & Objectives / Intervention Plan Info	Designer	Regulatory activities	Informed Consent	Accepted
Exclusion Zone	Stakeholder (Case Worker)	Regulatory activities/ Safe neighbourhoods	Privacy	Accepted after Moderation
Curfew Alert	Stakeholder (Case Worker)	Regulatory activities/ Safe neighbourhoods	Privacy	Accepted after Moderation
Activity Meter Progress of Actions that the young person is planning to do in agreement with case worker.	Designer	Regulatory activities	Autonomy	Accepted
Asset Info (the presentation of summary information about the young person)	Designer	Regulatory activities	Informed Consent/ Privacy	Rejected

team working on the project included social scientists, criminologists and computer scientists.

The co-design activities in the various workshops yielded a rich set of data including design and specification of features/functions of the MAYOT app. Table 1 provides an overview of the main features that were identified. The feature "Exclusion Zone" is used to illustrate the tension of incorporating value sensitive issues into the design process and the implications on building resilience in end-users. Currently, exclusion zones are presented to young people by their case workers using printed maps with the areas that are prohibited marked on them. The limitations imposed by existing approaches are further exacerbated by delay in being informed about any breaches. Case workers only hear about a breach after the event so the damage to the young person's current order has already happened. Further discussion in the workshops established that case workers viewed "nudging" young people through the use of phone alerts as an emerging feature/function of the MAYOT App in regard to the exclusion order scenario. Inputs from the co-design workshop with the case workers were then used to take this stakeholder sourced prompt to prepare a mockup of the functions for supporting "Exclusion Zone". This prompt was then used as input in co-design workshops with the young people. At this point, resilience factors being addressed by the design feature aimed to provide regulatory activities as well as addressing safe neighbourhood requirements. Continued reflection on the Exclusion Zone in the later workshops with young people created a strong reaction and further scenarios were generated, but this time value sensitive concerns arose from different stakeholders, such as fear of being constantly monitored. We see a prompt that originated from a stakeholder, was presented as a designer prompt, and then reflected on as a stakeholder prompt and then subsequently acquiring value sensitive features in the co-design space. One of the outcomes from this evaluation workshop suggests that if a stated goal of an intended system is to develop or engender some concept of resilience in end-users (i.e. not the system itself) then the importance of aligning resilience with value sensitive properties can influence the acceptance of a feature in the system.

Our next task is to offer a formal analysis of value-sensitive co-design, in order to show how its properties implement our thesis on the effect of values on resilience. In particular, our formal analysis focuses on providing the necessary machinery that can elaborate how technologies help or hinder moral values.

4. Abstract Value-Sensitive Co-Design Theory

Our formal treatment is based on Abstract Design Theory (ADT), initially developed from General Design Theory (Yoshikawa, 1981; Kikuchi and Nagasaka, 2002) and formalized in the information theory settings of Barwise and Seligman (1997), then fully formally explored in Kakuda and Kikuchi (2001a, 2001b); Kikuchi (2003). We present an extension of this model meant to cover the notion of Value-Sensitive Co-Design.

4.1. Design Space

We start by defining the elements of the *design space* (otherwise known as functional scheme), obtained as the functional combination of *theory* and *state space*. A theory is construed by elements such as *entities*, *entities concepts*, *abstract concept* and *theory classification.*

Definition 1 (Entities Set)

We denote by $tok(T)$ the set of all real entities, also known as tokens, available to actors in a design process for a theory T.

Definition 2 (Entities Concept)

We denote by $typ(T)$ the set of entities concepts, also known as types, generated by $tok(T)$ by abstracting from the material properties of tokens of T and preserving their attributes and functions. Types represent the attribute values of a set of functions.

The relation between tokens and types in a design theory is defined by a *theory classification*:

Definition 3 (Theory Classification)

A theory classification is a triple

$$TCla = \langle tok(Cla(T)), typ(T), \models_{Cla(T)} \rangle$$

such that $typ(T)$ is the set of entities concepts or types of T; $\models_{Cla(T)}$ is a relation of values instantiation and values abstraction and $tok(Cla(T))$ is the set of consistent partitions of $typ(T)$.[1]

By the above definition: $typ_T(t) = \{\tau \in typ(T) : t \models_{Cla(T)} \tau\}$ for $t \in tok(Cla(T))$; and $tok(Cla(T))(\tau) = \{t \in tok(Cla(T)) : t \models_{Cla(T)} \tau\}$ for $\tau \in typ(T)$. The full set of classification relations of a theory $\models_{Cla(T)}$ generates the extension of a set of instantiations and abstractions that defines an abstract concept.

Definition 4 (Abstract Concept)

An abstract concept τ is obtained by an entity concept classified according to the values attributed to the entity.

Definition 5 (State Space)

The interpretation process of a theory into the design space is given by a *state space*

$$S = \langle tok(S), typ(S), state(S)\rangle$$

where: $tok(S)$ is the whole set of entities that can be instantiated in a scenario by an actor; $typ(S)$ is the whole set of prompts that can be abstracted from S by an actor and $state(S)$ is a map between prompts and scenarios.

From the state space, the actor formulates the *event classification.*

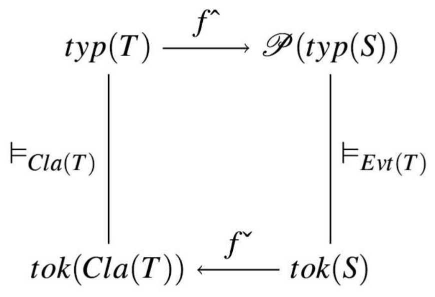
Definition 6 (Event Classification)

An event classification is a triple

$$ECla = \langle tok(S), \mathcal{P}(typ(S)), \models_{Evt(S)}\rangle$$

such that $tok(S)$ is a consistent set of events that can be instantiated by $typ(S)$; $\mathcal{P}(typ(S)) = \{A : A \subseteq typ(S)\}$, i.e. the power-set of attribute values entities concepts or types of S; $\models_{Evt(S)}$ is a relation of prompts instantiation and scenario events abstraction, such that $s \models_{Evt(S)} \sigma$ iff $\sigma \in state(S)$ for every $s \in tok(S)$.

By the functional combination of theory and state space, we define the *design space*, see Figure 1.



$$\begin{tikzcd}
typ(T) \arrow[r, "f^{\wedge}"] \arrow[d, dash, "\models_{Cla(T)}"'] & \mathscr{P}(typ(S)) \arrow[d, dash, "\models_{Evt(T)}"] \\
tok(Cla(T)) & tok(S) \arrow[l, "f^{\vee}"']
\end{tikzcd}$$

Figure 1. A single-actor design space

Definition 7 (Design Space)

A *design space* (also known as *functional scheme*) is a triple

$$DS = \langle T, S, f\rangle$$

where f is an infomorphism $f : TCla \leftrightarrows ECla$, i.e. a pair $f = \langle f^{\wedge}, f^{\vee}\rangle$ of maps $f^{\wedge} = typ(T) \rightarrow \mathcal{P}(typ(S))$ and $f^{\vee} = tok(S) \rightarrow tok(Cla(T))$ such that

$f^{\check{}}(s) \models_{Cla(T)} \tau$ iff $s \models_{Evt(S)} f^{\check{}}(\tau)$ for every $\tau \in typ(T)$ and $s \in tok(S)$. $f^{\hat{}}$ is called an interpretation of functions by values and thus maps from attributes values of conceptual entities to attribute values of entities in a state space; $f^{\check{}}$ is called an analysis about functions of entities and thus maps from a set of material entities in a state space to the set of consistent partitions of instantiable attribute values.

According to this functional model, a design space instantiates the design process: an actor starts by value based attributes in $typ(T)$, abstracted from real entities in the set $tok(Cla(T))$; when the actor selects such a value based attribute, it formulates it as an element $\pi \in \mathcal{P}(typ(S))$ in the design space; π is also called a *prompt* by the actor; a value-based prompt is then instantiated in an *event*, i.e. $s \in tok(S)$. If s instantiates a commuting diagram, it means it also satisfies the consistency requirement in $tok(Cla(T))$ and it becomes a *feature* of the design.

4.2. Co-Design Space

In this section we extend ADT to a more general model, where design is the result of an interaction of multiple actors. We start from two distinct design spaces, one for each of two actors, see Figure 2. We then establish formal constraints on their relations. These constraints are given in terms of Co-Theory Classification (Figure 3) and Co-Event Classification (Figure 4). We proceed now in defining such notions.

$$\begin{tikzcd}
typ(T_1) \arrow[r, "f_1^{\hat{}}"] \arrow[d, dash, "\models_{Cla(T_1)}"'] & \mathscr{P}(typ(S_1)) \arrow[d, dash, "\models_{Evt(T_1)}"] \\
tok(Cla(T_1)) & tok(S_1) \arrow[l, "f_1^{\check{}}"']
\end{tikzcd}$$

$$\begin{tikzcd}
typ(T_2) \arrow[r, "f_2^{\hat{}}"] \arrow[d, dash, "\models_{Cla(T_2)}"'] & \mathscr{P}(typ(S_2)) \arrow[d, dash, "\models_{Evt(T_2)}"] \\
tok(Cla(T_2)) & tok(S_2) \arrow[l, "f_2^{\check{}}"']
\end{tikzcd}$$

Figure 2. Design spaces for two distinct actors

Definition 8 (Co-Theory Classification)

A co-theory classification is a triple $Co - TCla = \langle TCla_1, TCla_2, g\rangle$ where $TCla_1$ and $TCla_2$ are theory classifications for distinct actors A_1, A_2 and g is an infomorphism $g = \langle g^{\wedge}, g^{\vee}\rangle$ of maps $g^{\wedge} = typ(T_1) \leftrightarrow typ(T_2)$ and $g^{\vee} = tok(Cla(T_1)) \leftrightarrow tok(Cla(T_2))$ such that

- $g^{\vee}(t_2) \models_{Cla(T_1)} \tau_1$ iff $t_2 \vdash_{Cla(T_2)} g^{\wedge}(\tau_1)$ for every $\tau_1 \in typ(T_1)$ and $t_2 \in tok(Cla(T_2))$, and
- $g^{\vee}(t_1) \models_{Cla(T_2)} \tau_2$ iff $t_1 \vdash_{Cla(T_1)} g^{\wedge}(\tau_2)$ for every $\tau_2 \in typ(T_2)$ and $t_1 \in tok(Cla(T_1))$.

$$\begin{tikzcd}
typ(T_1) \arrow[r, dash, "g^{\wedge}"] \arrow[d, dash, "\models_{Cla(T_1)}"'] & typ(T_2) \arrow[d, dash, "\models_{Cla(T_2)}"] \\
tok(Cla(T_1)) \arrow[r, dash, "g^{\vee}"'] & tok(Cla(T_2))
\end{tikzcd}$$

Figure 3. Co-theory classification for two actors

A co-theory classification includes therefore a (two-ways) *theory interpretation* $g^{\wedge} = typ(T_1) \leftrightarrow typ(T_2)$ between the two actors which holds iff for every realization of a token on a type of T_1, there exists a function that translates into a corresponding realization of a token on a type of T_2; and a (two-ways) *theory model* $g^{\vee} = tok(Cla(T_1)) \leftrightarrow tok(Cla(T_2))$ which holds iff for every instantiation of a type in token of T_1, there exists a function that translates into a corresponding instantiation of a type in a token of T_2.

Definition 9 (Co-Event Classification)

A co-event classification is a triple $Co - ECla = \langle ECla_1, ECla_2, h\rangle$ where $ECla_1$ and $ECla_2$ are event classifications for distinct actors A_1, A_2 and h is an infomorphism $h = \langle h^{\wedge}, h^{\vee}\rangle$ of maps $h^{\wedge} = \mathcal{P}(typ(S_1)) \leftrightarrow \mathcal{P}(typ(S_2))$ and $h^{\vee} = tok(S_1) \leftrightarrow tok(S_2)$ such that

- $h^{\vee}(s_2) \models_{Evt(T_1)} \pi_1$ iff $s_2 \models_{Evt(T_2)} h^{\wedge}(\pi_1)$ for every $\pi_1 \in \mathcal{P}(typ(S_1))$ and $s_2 \in tok(S_2)$, and
- $h^{\vee}(s_1) \models_{Evt(T_2)} \pi_2$ iff $s_1 \models_{Evt(T_1)} h^{\wedge}(\pi_2)$ for every $\pi_2 \in \mathcal{P}(typ(S_2))$ and $s_1 \in tok(S_1)$.

$$\begin{tikzcd}
\mathscr{P}(typ(S_1)) \arrow[r, dash, "h^{\wedge}"] \arrow[d, dash, "\models_{Evt(T_1)}"'] & \mathscr{P}(typ(S_2)) \arrow[d, dash, "\models_{Evt(T_2)}"] \\
tok(S_1) \arrow[r, dash, "h^{\vee}"'] & tok(S_2)
\end{tikzcd}$$

Figure 4. Co-event classification for two actors

A co-event classification includes therefore a (two-ways) *prompt interpretation* $h^{\wedge} = \mathcal{P}(typ(S_1)) \leftrightarrow \mathcal{P}(typ(S_2))$ between two actors which holds iff for every realization of an event on a prompt from S_1, there exists a function that translates into a corresponding realization of an event on a prompt from S_2; and a (two-ways) *event model* $h^{\vee} = tok(S_1) \leftrightarrow tok(S_2)$ between two actors which holds iff for every instantiation of a prompt in an event in S_1, there exists a function that translates into a corresponding instantiation of a prompt in an event in S_2.

The co-design space is completed by two more diagrams. To obtain the first, called Co-Theory Selection (Figure 5), we combine the theory and prompt interpretations from the two actors.

Definition 10 (Co-Theory Selection)

A co-theory selection is a tuple $Co - TSel = \langle f_1{}^{\wedge}, f_2{}^{\wedge}, g^{\wedge}, h^{\wedge} \rangle$ of maps

- $f_1{}^{\wedge}$: the *value interpretation* extracted from the individual design space of actor A_1;
- $f_2{}^{\wedge}$: the *value interpretation* extracted from the individual design space of actor A_2;
- $g^{\wedge}$: the *theory interpretation* extracted from the co-theory classification;
- $h^{\wedge}$: the *prompt interpretation* extracted from the co-event classification.

$$\begin{tikzcd}
typ(T_1) \arrow[r, "f_1{}^{\wedge}"] \arrow[d, dash, "g^{\wedge}"'] & \mathscr{P}(typ(S_1)) \arrow[d, dash, "h^{\wedge}"] \\
typ(T_2) \arrow[r, "f_2{}^{\wedge}"'] & \mathscr{P}(typ(S_2))
\end{tikzcd}$$

Figure 5. Co-theory selection for two actors

The last diagram, called Co-Event Selection (Figure 6), is obtained by combining the theory and event models from the two actors.

Definition 11 (Co-Event Selection)

A co-event selection is a tuple $Co - ESel = \langle f_1{}^{\check{}}, f_2{}^{\check{}}, g^{\check{}}, h^{\check{}} \rangle$ of maps

- $f_1{}^{\check{}}$: the *function analysis* extracted from the individual design space of actor A_1;
- $f_2{}^{\check{}}$: the *function analysis* extracted from the individual design space of actor A_2;
- $g^{\check{}}$: the *theory model* extracted from the co-theory classification;
- $h^{\check{}}$: the *event model* extracted from the co-event classification.

$$\begin{tikzcd}
tok(Cla(T_1)) \arrow[r, "f_1{}^{\check{}}"] \arrow[d, dash, "g^{\check{}}"'] & tok(S_1) \arrow[d, dash, "h^{\check{}}"] \\
tok(Cla(T_2)) \arrow[r, "f_2{}^{\check{}}"'] & tok(S_2)
\end{tikzcd}$$

Figure 6. Co-event selection for two actors

We can now finally define the formal notion of a co-design space (Figure 7):

Definition 12 (Co-Design Space)

A co-design space for two (or more) actors is a tuple

$$CD = \langle Co - TCla, Co - ECla, Co - TSel, Co - ESel \rangle$$

which includes a Co-Theory Classification, a Co-Event Classification, a Co-Theory Selection and a Co-Event selection.

A feature of a co-design space can be defined informally as a property of the space which satisfies theory (values) and events (interpretation) for each and all of the actors. This results formally in a pair of co-theory selection and co-event selection each satisfying the other:

$$\begin{tikzcd}
 & typ(T_2) \arrow[rr, dash, "f_2^{\wedge}"] \arrow[dl, dash, "g^{\wedge}"'] \arrow[dd, dotted, dash, "\models_{Cla(T_2)}" near start] & & \mathscr{P}(typ(S_2)) \arrow[dl, dash, "h^{\wedge}"] \arrow[dd, dash, "\models_{Evt(T_2)}"] \\
typ(T_1) \arrow[rr, dash, "f_1^{\wedge}"] \arrow[dd, dash, "\models_{Cla(T_1)}"'] & & \mathscr{P}(typ(S_1)) \arrow[dd, dash, "\models_{Evt(T_1)}" near start] & \\
 & tok(Cla(T_2)) \arrow[rr, dotted, dash, "f_2^{\vee}"] \arrow[dl, dotted, dash, "g^{\vee}"'] & & tok(S_2) \arrow[dl, dash, "h^{\vee}"] \\
tok(Cla(T_1)) \arrow[rr, dash, "f_1^{\vee}"'] & & tok(S_1) &
\end{tikzcd}$$

Figure 7. The co-design space

Definition 13 (Co-feature)

A feature m of a co-design space (also known as a co-feature) is a pair of mappings:

$$m = \langle (Co-TSel \models_{Evt(T_{i,j})} Co-ESel), (Co-ESel \models_{Cla(T_{i,j})} Co-TSel) \rangle$$

4.3. Examples

We now offer some examples of prompts within a co-design space, mimicking the co-workshops from our study-case in Section 3, in order to show how some become features and others do not. The initial situation before the workshop represents our starting point. A prompt for an *ExclusionZone* property is issued by a designer under a *regulation* value. Within the designer's space, the value is classified as a *restriction* function, the property is instantiated by a reminder event, instantiated by two functions *SMSsend* and *SMSget*, which represents the initial mechanism for implementing how a young person is advised of a potential violation of an *ExclusionZone*. In order this to become a feature in the co-design space, each element needs to map in the other stakeholder's space under appropriate model and value interpretation. For the interpretation: *regulation* maps to *privacy*, *ExclusionZone* to *AllowedZone*; for the model, the *restriction* classification maps to *information*, and the event *SMSreminder* is valid under the second space, in that it satifies the main value *privacy*. This accepted prompt is visualized in Figure 8.

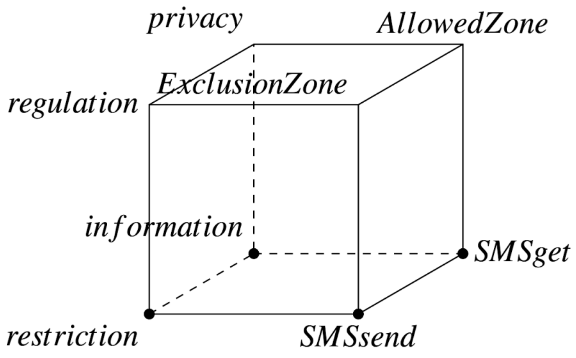


Figure 8. Accepted Prompt

In the second workshop, a new prompt is suggested by the designer. The previous event *SMSsend* is substituted by a *GeoTrack* event by map alerts which utilises the mapping and geo-positioning capabilities of a smart phone to report on a potential violation of an *ExclusionZone*. In the other stakeholder's space, this event is in conflict with the classification *information* that has to satisfy the *privacy* value, hence it is rejected and it does not become a feature. This result of the co-design space is reflected in Figure 9.

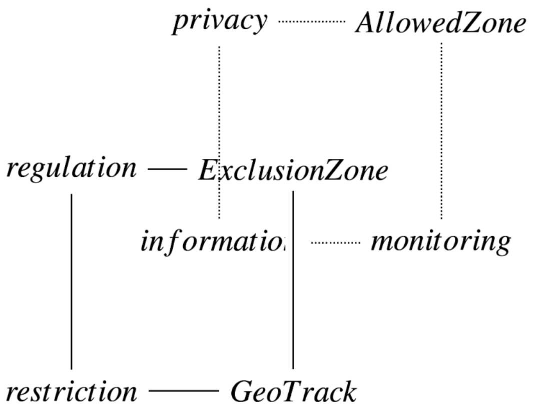


Figure 9. Rejected Prompt

In the actual co-design workshop related to the MAYOT project, this situation was resolved by a modification on the prompt: the option to opt-out from a pre-installed geo-tracking MAYOT app feature coupled with an alert was deemed sufficient to resolve the conflict, representing an option compatible with both privacy and regulatory values. We represent this modification with an additional *optout* node restoring commutativity in our graph, see Figure 10. In this modified graph, the new node *optout* creates a composition function from the previously un-mapped nodes *geotrack* and *monitoring*.

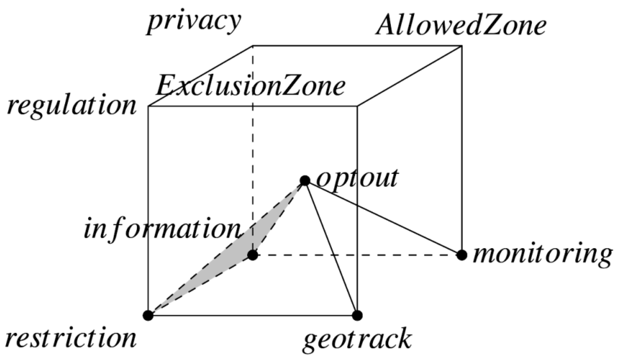


Figure 10. Extended Prompt

5. Resilient Co-Design

Our formal model of co-design offers working definitions of resilience. At a low-level of description, it refers to a resilient prompt:

Definition 14 (Resilient Prompt)

A prompt of an Information System is considered resilient if it is a feature in the system's current configuration which resists all value-based relations of currently involved stakeholders.

At the next level, we identify *resilience of a system*:

Definition 15 (Resilient System)

A system is resilient if its prompts remain features in view of future, possible, foreseeable value-based extensions of its current configuration. Such extensions can be understood in terms of new prompts from the current stakeholders, or as involving new stakeholders and are restricted in terms of the number of admissible threats to resilience.

Our analysis suggests that a feature in a value-based model of co-design presents a stronger level of resilience to further changes of the system. Accordingly, we offer the following definition

Definition 16 (Resilience for the End-User)

An Information System is *end-user resilient* if its future configurations are obtained by prompts that satisfy the users' theory classifications.

A design process that accounts for relevant values is bound to produce a system resilient for the end-users involved. Finally, the design process itself can be accounted as a *measure of the resilience* of a system:

Definition 17 (Measure of Resilience)

An Information System can be called *strongly resilient* if, in view of possibly conflicting value-based future configurations, it admits of new relations accommodating them.

A thorough comparison with current definitions of resilience from other disciplines is outside the scope of this paper, but our characterization offers a concrete basis to suggests that *value* – a largely ignored property in such studies – is a necessary definitional trait in resilience for information systems. Moreover, we also consider measure of such a property a valuable asset of the current study, requiring further exploration.

Theorem 1 (Resilient Co-Design)

A co-design space is resilient with respect to a new entity if and only if the latter allows infomorphisms of maps with the current theory and event classifications of all actors.

Proof.

The proof proceeds on the structure of the co-design space:

- $\rightarrow$: assume, by contradiction, that a co-design space is not resilient with a new prompted entity; then by Definition 14 of resilient prompt, such a prompted entity will not satisfy at least one value based relation of a currently involved stakeholder. By Definition 2, attributes value of functions are modelled in a design space as types $typ(T_1)$; a type proposed by a stakeholder, call it τ_1, will by definition be consistent in her Theory Classification, and if it does not satisfy another stakeholder's, the latter will have a token t_2 realised in the current design such that either $g^{\check{}}(t_2) \not\models_{Cla(T_1)} \tau_1$, or $g^{\check{}}(t_1) \not\models_{Cla(T_2)} \tau_2$. In either case, there is an invalid infomorphism with some of the current theory and classification events, contrary to the assumption.
- $\leftarrow$: assume infomorphisms of maps for all current theory and event classifications of all actors can be defined for a new entity τ prompted; then, by Definition 14, τ is a resilient prompt. $\square$

6. Conclusions

Besides the problem of conflicting values explicitly addressed by our formal theory, the source of prompts engendering values changes over time, and it is a necessary but difficult task to keep a trace of how values evolve, and how to express such changes in a formal setting. This suggests that it is not sufficient to draw attention to value concerns without continuing to explore how they then embed themselves in a wider software development process (Primiero et al., 2019). To this aim, one might ask what purpose can a framework for managing value sensitive concerns serve, and how does the framework enhance existing practices of analysis and design.

We suggest that developing a clear (in terms of formal modeling), causal, traceable link that chains together value concerns, stakeholders raising them, the scenarios and space in which this happens and the system's feature that is impacted, can provide a more nuanced computation for determining priorities and returns on investments found in more traditional requirements engineering approaches. A second benefit of tracking value concerns is the opportunity provided for early evaluation of a particular concern. Other research that we have undertaken examines how values related to privacy can be subject to early evaluation through the use of expert domain knowledge encoded in a Bayesian network (Barn et al., 2015). A final foreseeable advantage is the exploitation of results from information theory on channels and classifications from the underlying design theory, applied to the context of value-based design. The present contribution only offers the general framework in which such further analyses can be formulated, while at the same time it significantly extends on the content and applicability of the original theory from Kakuda and Kikuchi (2001b).

The participatory design school has had a traditional strength in examining the socio-technical elements of systems design, but artefacts from that approach are partially lost in translation in the task of implementation. We suggest that such a translation loss may be attributed to a lack of appropriate languages to bridge different domains. Our framework presents elements of a language design and relationships to software implementation concepts that may contribute to addressing this need. There are further stages to the language design process: specifically the need for tool support. The availability of formally defined notions and relations belonging to the design process is a crucial step for implementation in a programming language that could help automatizing the identification of value-based conflicts at the design stage. (Moral) values are closely linked to goals and it is tempting to extend existing languages for goal-oriented meta-models – such as KAOS (Darimont et al, 1997) and i* (Yu, 1997) – to include concepts that we have identified in this paper. We suggest that the relationship between socio-technical systems design and software implementation is significant and the route that we have taken (namely: from participatory to value-sensitive to formal design) is more relevant than that taken by goal oriented requirements engineering based approaches for the class of socio-technical system we have produced.

Resilience in end-users remains a zeitgeist phenomena that is relatively neglected in the IS methodology literature. Resilient systems depend upon resilience in their end-users. We have proposed that resilience is a composite structure, derived from human or moral values inherent in individuals. If an IS system can demonstrate through the design process how value conflicts are resolved and supported, it is ultimately better placed to engender resilience in individuals and hence to be regarded as a resilient IS. Currently, research in VSD whilst acknowledging and accounting for values in the design of systems has not addressed the core issue of value resolution. We have presented a formal model for value resolution using empirical data from the MAYOT project. The formal model, whilst providing a foundational basis for value sensitive co-design, brings with it limitations arising from perceived complexity from the designer's perspective. One future direction of our work will explore the use of alternative diagrammatic representations of the formal model, possibly by developing a domain specific language (e.g. as a Unified Modelling Language (UML) profile, Rumbaugh et al. (2004)). Such an approach will also lend itself to software tools for supporting the co-design process.

Acknowledgements

Special thanks to the case workers and the young people who participated in the co-design workshops. Early results from this research were presented at the Conference of the International Association of Computing and Philosophy, and in its latest version at the I-CORE "Complexity, Inter-Connectivity and Resilience" at the University of Bologna in 2019. Thanks to participants to these events for feedback, and to Carolyn Budd, Lalith Kaushik and Franco Raimondi for their input. This research was supported in part by the Nominet Trust, UK, by the Project PROGRAMme ANR-17-CE38-0003-01, and by the Department of Philosophy "Piero Martinetti" of the University of Milan under the project "Departments of Excellence 2018–2022" awarded by the Ministry of Education, University and Research (MIUR).

NOTE

[1] By imposing that $tok(Cla(T))$ is the set of consistent partitions, both equivalent and empty subsets are not taken into account.

REFERENCES

Alcoff, L. (1991) The problem of speaking for others, *Cultural critique*, n. 20, pp. 5–32.

Barn, B., Primiero, G. and Barn, R. (2015) An approach to early evaluation of informational privacy requirements. SAC '15: Proceedings of the 30th Annual ACM Symposium on Applied Computing, April 2015, pages 1370–1375, https://doi.org/10.1145/2695664.2695788

Barwise J., Seligman, J. (1997) *Information Flow: the logic of distributed systems.* Cambridge University Press.

Beyer H., Holtzblatt K. (1998) *Contextual design: defining customer-centered systems.* Morgan Kaufmann Pub.

Bjerknes G., Ehn P., Kyng M., Nygaard K. (1987) *Computers and democracy: A Scandinavian challenge.* Gower Pub Co.

Borning, A. and Muller, M. (2012) Next steps for value sensitive design, *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pp. 1125–1134.

Butschi, D., Courant, M. and Hilty, L.M. (2005) Towards sustainable pervasive computing – Guest editorial, *IEEE Technology and Society Magazine*, vol. 24(1), pp. 7–8.

Callister W, (1994) Mechanical properties of metals. In *Materials science and engineering, an introduction*, New York: John Wiley & Sons, Inc, pp. 106–147.

Colburn, T. R., (1999). Software, Abstraction, and Ontology, *The Monist*, 82(1): 3-19.

Darimont, R., Delor, E., Massonet, P., van Lamsweerde, A. (1997) GRAIL/KAOS: an environment for goal-driven requirements engineering, *Proceedings of the 19th ACM international conference on Software engineering*, pp. 612–613.

Davis F.D. (1993) User acceptance of information technology: system characteristics, user perceptions and behavioral impacts, *International Journal of Man-Machine Studies*, vol. 38, pp. 475–487.

Eisenhardt, Kathleen M. (1989) Building theories from case study research, *Academy of management review*, vol. 14(4), pp. 532–550.

Erol O., Sauser B.J., Mansouri M. (2010) A framework for investigation into extended enterprise resilience. *Enterprise Information Systems* 4(2): 111–136.

Friedman B., Kahn Jr P.H., Borning A. (2006) Value sensitive design and information systems. *Human-computer interaction in management information systems: Foundations* 5: 348–372.

Gao H., Barbier G., Goolsby R. (2011) Harnessing the crowdsourcing power of social media for disaster relief. *Intelligent Systems, IEEE* 26(3): 10–14.

Greenbaum J., Kyng M. (1991) Design at work: Cooperative design of computer systems. L. Erlbaum Associates Inc. Hillsdale, NJ, USA.

Gellman, B. (2014) Snowden has shown the 'huge disparity of surveillance and power'. *The Guardian*, 10/03/2014, http://www.theguardian.com/technology /2014/mar/10/edward-snowden-surveillance-government-nsa-gchq-barton-gellman?CMP=twt_gu

Greenberg M.T. (2006) Promoting resilience in children and youth. *Annals of the New York Academy of Sciences* 1094(1): 139–150.

Holling C.S. (1973) Resilience and stability of ecological systems. *Annual review of ecology and systematics* 4: 1–23.

Hughes A.L., Palen L. (2009) Twitter adoption and use in mass convergence and emergency events. *International Journal of Emergency Management* 6(3): 248–260.

Hunter C. (2012) Is resilience still a useful concept when working with children and young people? *Australian Institute of Family Studies*, CFCA Paper No. 2, April 2012.

Kakuda Y., Kikuchi M. (2001a) Topology on classifications in abstract design theory. *Proc. IWES'01*, pp. 123–130.

Kakuda Y., Kikuchi M. (2001b) Abstract design theory. *Annals Japan Association of Philosophy of Science* 29(1): 1–6.

Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., Carriere, J. (1998) The architecture tradeoff analysis method, *Engineering of Complex Computer Systems, 1998. ICECCS'98. Proceedings. Fourth IEEE International Conference on*, pp. 68–78.

Kikuchi M. (2003) Analysis and design from a viewpoint of information flow. In: Löwe B., Malzkom W., Räsch T. (eds) *Foundations of the Formal Sciences II*, Trends in Logic, vol. 17, Springer Netherlands, pp. 119–122

Kikuchi M., Nagasaka I. (2002) On the three axioms of general design theory. *Proc. IWES'02*, pp. 69–76.

Kujala S., Väänänen-Vainio-Mattila K. (2009) Value of information systems and products: Understanding the users' perspective and values. *Journal of Information Technology Theory and Application* 9(4): 23–39.

Mark G.J., Al-Ani B., Semaan B. (2009) Resilience through technology adoption: merging the old and the new in Iraq. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM*, pp. 689–698.

MoJ (2013) Youth Justice Statistics 2011/12 England and Wales. Ministry of Justice.

Moor J.H. (1978). Three Myths of Computer Science, *The British Journal for the Philosophy of Science*, 29(3): 213-222.

Nellis M. (2004) The 'Tracking' Controversy: The Roots of Mentoring and Electronic Monitoring. *Youth Justice* 4(2): 77.

OECD (2008) Concepts and dilemmas of state building in fragile situations: From fragility to resilience. *Journal on Development* Volume 9, No. 3.

OECD (2013) *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*, Organisation for Economic Co-operation and Development.

Primiero, G., (2016), Information in the philosophy of computer science, in Floridi L. (ed) *The Routledge Handbook of Philosophy of Information*, 90–106, Routledge.

Primiero, G., Raimondi, F., Chen, T., (2019). A theory of change for prioritised resilient and evolvable software systems, *Synthese*, doi:10.1007/s11229-019-02305-7.

Primiero, G., (2020), *On the Foundations of Computing*, Oxford University Press.

Rumbaugh, James and Jacobson, Ivar and Booch, Grady (2004). *The Unified Modeling Language Reference Manual*, Pearson Higher Education.

Rutter M. (2006) Implications of resilience concepts for scientific understanding. *Annals of the New York Academy of Sciences* 1094(1): 1–12.

Sanders E.N. (2000) Generative tools for co-designing. In: *Collaborative design*, Springer, pp. 3–12.

Schön D.A. (1983) *The reflective practitioner: How professionals think in action*, vol. 5126. Basic books.

Shklovski I., Palen L., Sutton J. (2008) Finding community through information and communication technology in disaster response. *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, ACM, pp. 127–136.

Smith D., Goldson B., Muncie J. (2006) Youth crime and justice: research, evaluation and 'evidence'. *Youth, crime and justice: critical issues*, p. 78.

Turner, R., (2011). Specification, *Minds & Machines*, 21(2): 135–152.

Ungar M. (2013) Resilience, trauma, context, and culture. *Trauma, Violence, & Abuse* 14(3): 255–266.

Venkatesh V., Morris M.G., Davis G.B., Davis F.D. (2003) User acceptance of information technology: Toward a unified view. *MIS quarterly*, pp. 425–478.

Wright M.O., Masten A.S., Narayan A.J. (2013) Resilience processes in development: Four waves of research on positive adaptation in the context of adversity. In: *Handbook of Resilience in Children*, Springer, pp. 15–37.

Yin, R. K. (2009) *Case study research: Design and methods* (4th Ed.). Thousand Oaks, CA: Sage.

Yoo D., Huldtgren A., Woelfer J.P., Hendry D.G., Friedman B. (2013) A value sensitive action-reflection model: evolving a co-design space with stakeholder and designer prompts. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, pp. 419–428.

Yoshikawa H. (1981) General Design Theory and a CAD system, In: *Man-machine Communication in CAD/CAM*, T. Sata, E. Warman (editors), pp. 35–57, North-Holland, Amsterdam.

Yu E.S.K. (1991) Towards modelling and reasoning support for early-phase requirements engineering, in *Requirements Engineering, 1997, Proceedings of the Third IEEE International Symposium on*, pp. 226-235.