

Values@Runtime: An Adaptive Framework for Operationalising Values

Amel Bennaceur*, Diane Hassett†, Bashar Nuseibeh*, and Andrea Zisman*

*The Open University, UK

firstname.lastname@open.ac.uk

†Lero, Ireland

firstname.lastname@lero.ie

Abstract—We present an adaptive framework to assist users in making more value-sensitive decisions during their (runtime) use of software. The framework enables users to (i) represent, instantiate, and monitor their values and behaviour; (ii) understand mismatches between stated values and their observed behaviour; and (iii) recommend ways to align users’ values and behaviour. We built a *values shopping basket* tool to illustrate and demonstrate the adaptive framework in the food consumption domain, a sector that is rich in values and regularly undergoes reflection and debate.

Index Terms—Values@Runtime, operationalisation, adaptation, recommendations, reflection

General Abstract - Society is pondering the values it cherishes, and users increasingly find themselves reflecting on which values are important to them. With software playing a crucial role in society and having a significant impact on how we live, the way we engineer and use software must take into account those values. In this paper we present a framework to support users to articulate, measure, and reflect on their values as they interact with software systems. The rationale is that users gain better understanding of their values as they experience, reflect and learn about them, when making decisions mediated by software. We demonstrate our framework through a *values shopping basket* prototype that enables users to specify, reflect, and make value-sensitive decisions during food purchase.

I. INTRODUCTION

Recent work has promoted the need to consider ethics [1] and values [2] during the development of software systems. As outlined by Mougouei *et al.* [3] “people are demanding that practitioners align technologies with human values”. Some approaches have been proposed to assess and study values in software engineering [4], to incorporate social values in software design patterns [5], and to measure the impact of values in requirements engineering activities [6]. Values are well studied in human-computer interaction and information systems [7] [8]. Existing approaches focus on early stages of the development process [9], with little attention given to the satisfaction of values in deployed software systems.

In this paper we complement existing approaches by considering values during runtime operation of software systems in general, and user-centric software in particular. We seek to show how software can help users articulate, measure and

reflect on their values at runtime. It is common for users to gain better understanding of their values as they experience, reflect and learn more about them [10].

We propose the notion of *Values@Runtime*, in which values are better understood at runtime, building on the concepts of Models@Runtime [11] and Requirements@Runtime [12] that aim to deal with uncertainty by delaying some decisions until software is in operation. In *Values@Runtime* adaptation is configured for and together with users rather than by the software system alone. The adaptive process aims to engage users and to support learning about models of users’ values.

We propose an adaptive framework for operationalising values during runtime in user-centric applications. In this framework we provide values instantiation as a means of representing the concrete actions that users associate with values [13]. Our adaptive framework supports values operationalisation in terms of (i) representation, instantiation, and monitoring of values and behaviour; (ii) understanding existing mismatches between values and users’ behaviour based on analysis; and (iii) recommending ways to align values and behaviour as well as reflecting on the recommendations.

The operationalisation characteristics of our work were influenced by using a prototype software *values shopping basket* that we developed to assist users in making value-sensitive decisions at runtime. We investigate mismatches between values stated by users and how users behave relative to those values in the food consumption domain; a domain rich in values, and which regularly undergoes reflection and debate.

Our framework seeks to achieve the following objectives: (1) refine values descriptions into domain-specific attributes to enable reasoning about them; (2) instantiate values based on users’ understanding; (3) monitor users’ behaviour with respect to instantiated values; (4) identify mismatches between instantiated values and users’ behaviour; (5) support values and behaviour alignment through recommendations; and (6) reflect on recommendations.

The remainder of the paper is structured as follows. Section II reviews related work. Section III presents our adaptive framework for operationalising values during runtime. Section IV introduces our case study of a values shopping basket. Finally, Section V discusses the work and future directions.

II. BACKGROUND AND RELATED WORK

The role of values in the design of software technology has received significant recent attention [9]. A thorough review is beyond the scope of this short paper. We summarise below the most relevant work.

Human values denote life goals which establish what is important to a person [14]. Schwartz identified ten universal values categories organised to reflect congruence and conflict between values. Maio *et al.* [15] develop values from abstract concepts to more concrete instantiations by associating values, attitudes and behaviours. They suggest values as mental representations which can be studied at three interconnected levels: (i) the system level, by which values are connected to each other (Schwartz’s model); (ii) the abstract level, which comprises the importance that people attach to the abstract concepts; and (iii) the instantiation level, which includes specific situations, issues, and behaviors relevant to values [15]. We build on this work to identify mismatches and align stated values with observed behaviour.

In **value-based requirements engineering** [16] values are seen as personal attitudes and beliefs which influence functional and non-functional requirements. Human values are treated as software requirements, specifically as softgoals or non-functional requirements [17]. Ferrario *et al.* [18] argue that complex ‘wicked’ problems such as sustainability should be treated as softgoals, not as functional requirements.

Several approaches promote the importance of considering human **values during the development of software systems** and we refer interested readers to the comprehensive recent survey paper [9]. The authors highlight that existing approaches support values operationalisation in early stages of software development (requirements and design), but later stages (implementation and testing) still need attention.

In **values-first software engineering**, developers are prompted to “*understand the values driving SE and encourage values reflection before attempting to operationalise human values*” [18]. Winter *et al.* [4] highlight the need to study values at the instantiation level, where values are linked to behaviour, and propose the use of *Values Q-sort* [4], which requires participants to rank statements based on their agreement with them. Their study highlights how, at design time, value instantiations are hard to elicit, and become often generic and abstract, only providing insight into espoused attitudes and behaviours, and not indicative of the actual behaviour. This paper supports the instantiation level and proposes an adaptive framework where values are continuously articulated, measured, reflected upon, and refined (*Values@Runtime*). The rationale is from the fact that values are better understood *in situ*, i.e., when users are making value-sensitive choices.

In **value sensitive design** [19] researchers identify values of ethical importance using scenarios and storyboarding to elicit user attitudes and feelings during design processes. A framework of conceptual, empirical and technical investigations is used to help determine and elicit values, translate them into design requirements, and then prototype for feedback

and technical evaluation. Asikis *et al.* [20] propose a value-sensitive design approach that considers users’ sustainability preferences and match them with consumption products.

The adaptive framework we propose complements the notion of operationalising values in software by providing alignment and reflection of users’ values while users are engaging with the software. It measures user behaviour according to their stated values, allowing users to become aware of their own values. An exploratory study focusing on eliciting human values of general health and eHealth apps end users [21] found that domain context is important for values elicitation tools.

Applications now exist to support users purchasing sustainable food [22] and managing their diet and lifestyle [8]. They also support businesses and producers to manage supply and demand [23] and to provide an efficient and controlled production [24]. Although users may have **values that influence their food choices**, they may be unable to act due to lack of adequate information. Even though consumers are becoming better informed about the many issues that can arise in the food supply and demand chain, they may not be able to access specific information sources with the power to inform value-based purchasing behaviour. The Good Shopping Guide [25] and Ethical Consumer [26] provide online tools to support users to meet their values in their shopping experiences. However, they do not support the provision of values related information that is specific to a particular user/consumer.

The adaptive framework for *Values@Runtime* proposed in this paper overcomes the above drawbacks by providing a framework in which users instantiate values based on their understanding and become aware of when they are fulfilling or contradicting their values during product consumption.

III. VALUE ADAPTIVE FRAMEWORK

Operationalising values is defined as “*the process of identifying human values and translating them to accessible and concrete concepts so that they can be implemented, validated, verified, and measured in software*” [9]. Our framework complements the above definition to represent, monitor, match, align, recommend, and reflect on values during runtime operation of user-centric software systems.

We focus on systems where users make choices based on their values. Those choices involve products characterised by attributes that can be linked to values. Our operationalisation of values follows the MAPE-K loop [27], which iterates through monitoring (M), analysis (A), planning (P), and execution (E) over a knowledge base (K), to adapt the response of the system to information gathered at runtime.

Figure 1 shows an overview of the framework, with its main functions represented in blue and the data generated and used by these functions represented by arrows. The framework monitors the observed behaviour of users with respect to their stated values and product choices ❶. Values are then analysed in order to identify any mismatches between the values stated by the user and their behaviour ❷. When a mismatch is identified, the framework informs the user about the discrepancy and recommends substitutions of products that

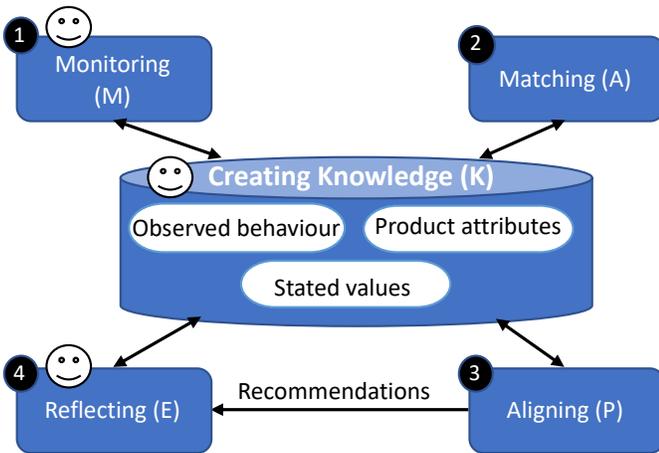


Fig. 1: Adaptive framework for operationalising values

are more aligned with the stated values ③. The choices of products and product substitutions are presented to the users as recommendations to enable users to reflect on their behaviour regarding their values ④. Reflection enables users to articulate their values better in future system iterations.

Creating Knowledge. Values can be described at different levels namely: system, abstract, or instantiation [15]. We complement this notion and advocate that value operationalisation requires finer grained descriptions of values (i.e., *refinement*) in terms of attributes characterising them. These attributes need to be *instantiated* with concrete content to support measuring values satisfiability. For example, *sustainability* is a high level value that needs to be refined by attributes such as seasonal, organic, animal welfare, distance of travel, locally-sourced, or renewability. Each of these attributes needs to be instantiated depending on the user and the situation. Locally-sourced food may mean an item produced in the same region to one user, and in the same country to another; distance of travel may be 500 miles at most for one user, and 12,000 miles for another.

The finer grained descriptions of values and their instantiations are grounded to *specific domains*. Sustainability in the food domain may relate to seasonality and animal welfare, while in the energy domain it may relate to renewability. The types of attributes characterising values in a domain may also *vary* from one user to another. For example, sustainability in the food domain may mean seasonal and locally-sourced food to one user, but it may mean animal welfare and organic production to another user. Within a domain, the attributes characterising a value for a user may also *change over time* depending on the user's experience, reflection, and learning. For a user, sustainability in the food domain may relate to seasonality and animal welfare. However, as the user makes decisions about food products, s/he may realise that the source of a product becomes an important parameter and, therefore, locally-sourced food should be added as an attribute to characterise sustainability for this user.

The attributes and instantiations of values are also *context-*

dependent and *product-dependent*. As an example of context-dependent value, a user may have different preferences when eating at home and when eating at a restaurant: a vegan user may eat vegetarian if eating with friends at a restaurant. As an example of product-dependent value, a user may only consume locally-sourced products overall, but is prepared to consume wine from different parts of the world.

To relate users' values to available products it is also necessary to describe products using similar, but not necessarily the same, attributes. These attributes are also domain-dependent, may vary among products, and may change over time. A food product needs to hold information about its availability, organic level, distance of travel, or location of production. A user may define sustainability in terms of seasonality, but in a product this is characterised by the product being in season.

Monitoring and Matching. Existing monitoring approaches support observation of well-defined system requirements and how they are satisfied by the system [28]. However, it is necessary to develop new value-sensitive monitoring techniques that consider the behaviour of users, or group of users, with respect to their values. The results of the monitoring activity may identify new values associated with the users.

In our adaptive framework, the stated values and observed behaviour are used in the matching and alignment activities. The observed behaviour can also influence changes in the descriptions of stated values (attributes and instantiations). The framework supports matching of (i) users' values and products' attributes, and (ii) values stated by the users and observed users' behaviour with the system.

Given that several attributes and measures are involved in the refinement and instantiation of users' values and the attributes of products, and that some of those attributes may not exist for specific products, matching stated values with products' attributes and matching stated values and observed behaviour needs to support the challenges of dealing with multiple criteria, uncertainty, and incompleteness.

The calculation of matching may be different for distinct types of values. For example, the matching for a value type may depend on the existence of one of its attributes (e.g., non-dairy products), while for another value type the matching may be calculated by the average scores of several attributes (e.g., animal welfare and distance of travel). Mismatches or conflicts may arise between stated values and observed behaviour. A user may instantiate sustainability in terms of seasonality of products, but may still buy/eat some products out of season.

Aligning and Reflecting. In order to deal with the mismatches or conflicts between stated values and observed behaviour, the framework provides recommendations to users. These recommendations range from information about how the stated values conflict with users' behaviour, to lists of products that better match the stated values of the users, and information about how the users can adapt their stated values.

The recommendations consider the matching of products with users' values and are based on earlier work [29] to

provide alternatives for users to *satisfice* and align their values. The aim is to provide users with values-sensitive insights into their behaviour at the micro-level i.e., when they are making choices about products; and at the macro-level, i.e., when they are making choices over multiple products. At the micro-level, users are alerted and made aware of any deviations from their stated values while they are making the choice. At the macro-level, users are invited to reflect on patterns of their behaviour to consider revising how they articulate their values. Recommendations are used to engage users to have control, set their own goals, and make their own decisions, which is paramount for software supporting behavioural changes [30].

IV. VALUES SHOPPING BASKET: A CASE STUDY

In order to illustrate our framework we developed a *values shopping basket* (VSB) tool. We used VSB to assist us in identifying the characteristics of an adaptive framework for operationalising values during runtime, and as a proof of concept to illustrate the approach. We implemented VSB as a mobile application to support food purchase. In the following, we illustrate the functions of the framework in this context.

Creating Knowledge. Although there are several values relevant to the food domain, we built upon the Food Ethics Council Framework [31] and selected three types of values, namely: *sustainability*, *culture*, and *social good*. We refined each of these values into several attributes, as illustrated in Table I. These attributes are not exhaustive and were selected due to available information in existing food products. The tool supports users to instantiate attributes of values in different ways. For example, in the case of culture, users specify if the ingredients in a product are part of their diet. For sustainability and social good, users are requested to score the importance of each attribute associated with the value in a 10-point Likert scale, or provide specific measures for the attributes when applicable (e.g., distance \leq 1500 miles). Users can also specify some attributes as *hard constraints*.

TABLE I: Exemplar Values and Associated Attributes

Value	Example Attributes
Culture	Vegetarian, Kosher, Halal, Pescatarian, No beef, No caffeine
Sustainability	Organic, Recyclable packaging, Travel distance
Social good	Fair trade, Rainforest Alliance certificate, Charity funding

Monitoring and Matching. The matching of values with products is based on scores measuring the satisfiability of the attributes of products with respect to the attributes associated with users' values. This satisfiability is calculated in different ways depending on the values type. For culture, the satisfiability is calculated based on the presence/absence of ingredients specified by a user. For sustainability and social good, the satisfiability is calculated by the weighted average of users' scores in the 10-point Likert scale. The observed behaviour of users is inferred by monitoring the history of purchased products and how these products satisfy stated values.

Aligning and Reflecting. In the tool, the alignment of values is supported by (i) informing the users how a purchased

product satisfies or conflicts with their values, and (ii) recommending alternative products that better match with the user's values. To inform the user, VSB displays the satisfiability scores of a product and all the history of purchases of a user. To provide users with a list of recommended products that better match their stated values, the tool carries out a search for product attributes that maximise the satisfiability scores associated with sustainability, culture, and social good. We formulated this search as a Multi-Objective Constrained Optimisation Problem [32]. A solution to this optimisation problem is a list of products whose attributes maximise the satisfiability scores of the stated values.

The users can reflect on the recommendations from the tool and gain insights into their behaviour. Users can respond by: (i) doing nothing and continuing to satisfy or conflict their values, (ii) following the recommendations and behaving in a way that aligns with their values (e.g., stop purchasing products that are not locally-sourced); or (iii) updating the description and understanding of their values (e.g., instantiating a different measure for distance of travel). The tool has demonstrated several characteristics of the adaptive framework. Currently, we are expanding the *values shopping basket* to support other types of values and to be used in different application domains.

V. DISCUSSION AND FUTURE WORK

The framework we propose in this paper aims to enable users to articulate and reflect on their values, by highlighting the alignment or mismatches between those values and users' behaviour while making choices. Going forward is also intended to provide software engineers with a concrete use case from which to value-aware software development processes and architectural patterns can be derived.

Conflicts & Tradeoffs. The current version of the framework specifically measures values of sustainability, culture and social good, but we acknowledge that there may exist conflicts with other types of values and other stakeholders, as well as (un)conscious value tradeoffs by the users. Conflicts can arise due to prioritisation of other values which can challenge compliance to stated values (e.g., sustainability vs. power) [14]. Other factors such as cost and availability may contribute to user observed behaviour. The framework supports users in reflecting about their choices in relation to their values while they are responsible for making those choices. These choices will contribute to creating new knowledge and supporting users to refine their values. We plan to explore further how to engage users in refining those values.

Users & Organisations. While we focused on users' values, there may exist conflicts with other stakeholders such as (food) organisations. These organisations maybe unable to meet the instantiated values or the attributes desired by the users in their products, or have different values from the users (e.g., profit). In these cases, the framework may empower users to act on their values and decide on the organisations with which to interact. We will extend the framework to incorporate values

of organisations and reason about conflicting values between users and organisations.

Uncertainty. The framework relies on available data about products' attributes for value instantiations [15]. Assuming that we represent values with relevant products' attributes at design time, these attributes and how they are associated with specific values will evolve. The framework will need to respond to incomplete or missing data. We anticipate *Values@Runtime* content to evolve based on users' input to modify and add other attributes associated with the product as more knowledge is created. It may be possible to extract some attributes automatically, or to obtain these attributes as a result of users' input. We plan to establish and build knowledge on how these attributes change to deal with uncertainty.

Mapping values and behaviour. The proposed framework aims to help users reflect on the alignment of their stated values and their behaviour by making those values explicit, and observing and recording value choices. Users can reflect on the values and behaviour and take appropriate decisions. They can restate their values, change their behaviour, or recognise situations in which mismatches between stated values and behaviour are acceptable. The framework aims to raise users' awareness and help them reflect on their stated values and behaviour, while keeping their agency and deciding whether to change or maintain their stated values or behaviour.

Generalisation. Values remain subjective and instantiating values requires domain knowledge and specificity. We are investigating the use of the framework in other domains such as values in the workplace. We welcome participation and inquiry from users, researchers, and other stakeholders within and beyond the boundaries of the food domain.

ACKNOWLEDGEMENTS

This work is supported by EPSRC Platform Grant on Secure Adaptable Usable Software Engineering (EP/R013144/1), UKRI Trustworthy Autonomous Systems Node in Resilience (EP/V026747/1), Science Foundation Ireland grants 16/RC/3918 (Confirm), and 13/RC/2094 P2 (Lero). We also thank Hakan E. Kiziloz for his input.

REFERENCES

- [1] A. Rashid, J. Weckert, and R. Lucas, "Software engineering ethics in a digital world," *IEEE Computer*, 2019.
- [2] J. Whittle, "Is your software valueless?" *IEEE Software*, 2019.
- [3] D. Mougouei, H. Perera, W. Hussain, R. Shams, and J. Whittle, "Operationalizing human values in software: A research roadmap," in *26th ACM Joint Meeting on European Soft. Engineering Conference and Symp. on the Foundations of Software Engineering*, 2018, pp. 780–784.
- [4] E. Winter, S. Forshaw, L. Hunt, and M. A. Ferrario, "Advancing the study of human values in software engineering," in *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 2019, pp. 19–26.
- [5] W. Hussain, D. Mougouei, and J. Whittle, "Integrating social values into software design patterns," in *International Workshop on Software Fairness*. ACM/IEEE, 2018.
- [6] H. Perera, R. Hoda, R. Shams, A. Nurwidyantoro, M. Shahin, W. Hussain, and J. Whittle, "The impact of considering human values during requirements engineering activities," *IEEE Transactions on Software Engineering*, 2021.
- [7] G. Cockton, "Value-centric HCI," in *NordHCI*, 2004.
- [8] C. Heath, C. Crivellaro, and L. Coles-Kemp, "Relations are more than bytes: Re-thinking the benefits of smart services with people and things," in *CHI Conf. on Human Factors in Computing Systems*. ACM, 2019.
- [9] M. Shahin, W. Hussain, A. Nurwidyantoro, H. Perera, R. A. Shams, J. C. Grundy, and J. Whittle, "Operationalizing human values in software engineering: A survey," *ArXiv*, vol. abs/2108.05624, 2021.
- [10] M. C. Gentile, *Giving voice to values*. Yale University Press, 2010.
- [11] N. Bencomo, S. Götz, and H. Song, "Models@run.time: a guided tour of the state of the art and research challenges," *Software & Systems Modeling*, vol. 18, 10 2019.
- [12] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. C. W. Finkelstein, "Requirements-aware systems: A research agenda for re for self-adaptive systems," *2010 18th IEEE International Requirements Engineering Conference*, pp. 95–103, 2010.
- [13] P. H. Hanel, K. C. Vione, U. Hahn, and G. R. Maio, "Value instantiations: the missing link between values and behavior?" in *Values and behavior*. Springer, 2017, pp. 175–190.
- [14] S. H. Schwartz, "An overview of the schwartz theory of basic values," *Online readings in Psychology and Culture*, vol. 2, no. 1, pp. 2307–0919, 2012.
- [15] G. R. Maio, *The psychology of human values*. Routledge, 2016.
- [16] S. Thew and A. Sutcliffe, "Value-based requirements engineering: method and experience," *Requirements engineering*, vol. 23, no. 4, pp. 443–464, 2018.
- [17] B. S. Barn, "Do you own a volkswagen? values as non-functional requirements," in *Human-Centered and Error-Resilient Systems Development*. Springer, 2016, pp. 151–162.
- [18] M. A. Ferrario, W. Simm, S. Forshaw, A. Gradinar, M. T. Smith, and I. Smith, "Values-first se: research principles in practice," in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 2016, pp. 553–562.
- [19] B. Friedman, P. H. Kahn, A. Borning, and A. Hultgren, "Value sensitive design and information systems," in *Early engagement and new technologies: Opening up the laboratory*. Springer, 2013, pp. 55–95.
- [20] T. Asikis, J. Klinglmayr, D. Helbing, and E. Pournaras, "How value-sensitive design can empower sustainable consumption," *Royal Society open science*, vol. 8, no. 1, p. 201418, 2021.
- [21] H. O. Obie, W. Hussain, X. Xia, J. Grundy, L. Li, B. Turhan, J. Whittle, and M. Shahin, "A first look at human values-violation in app reviews," in *43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 2021, pp. 29–38.
- [22] T. Asikis, J. Klinglmayr, D. Helbing, and E. Pournaras, "How value-sensitive design can empower sustainable consumption," *Royal Society Open Science*, vol. 8, no. 1, p. 201418, 2020.
- [23] S. Brewer, S. Pearson, L. Bidaut, J. G. Frey, G. Parr, and A. Zisman, "The internet of food things: adding value to the digitalisation of the UK food supply chain," *IET Conference Proceedings*, pp. 1–3, 2019.
- [24] V. Marinoudi, S. Claus, S. Pearson, and D. Bochtis, "Robotics and labour in agriculture. a context consideration," *Biosystems Engineering*, vol. 184, 2019.
- [25] "The good shopping guide," <https://thegoodshoppingguide.com/>.
- [26] "Ethical consumer," <https://www.ethicalconsumer.org/>.
- [27] IBM, "An architectural blueprint for autonomic computing," IBM, Tech. Rep., 2003.
- [28] R. Rabiser, M. Vierhauser, and P. Grünbacher, "Assessing the usefulness of a requirements monitoring tool: a study involving industrial software engineers," in *38th International Conference on Software Engineering, ICSE*. ACM, 2016, pp. 122–131.
- [29] A. Bennaceur, A. Zisman, C. McCormick, D. Barthaud, and B. Nuseibeh, "Won't take no for an answer: resource-driven requirements adaptation," in *14th International Symposium on Soft. Eng. for Adaptive and Self-Managing Systems, SEAMS@ICSE*. ACM, 2019, pp. 77–88.
- [30] I. Nahum-Shani, S. N. Smith, B. J. Spring, L. M. Collins, K. Witkiewitz, A. Tewari, and S. A. Murphy, "Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support," *Annals of Behavioral Medicine*, vol. 52, no. 6, pp. 446–462, 2018.
- [31] A. T. Höglund, "What shall we eat? an ethical framework for well-grounded food choices," *Journal of Agricultural and Environmental Ethics*, vol. 33, no. 2, pp. 283–297, 2020.
- [32] F. Rossi, P. Van Beek, and T. Walsh, *Handbook of constraint programming*. Elsevier Science, 2006.