UMI: Universal Modular Intervention Framework to Accelerate Behavior Change Support Systems Research

Tatsuya Yamaguchi[1,*], Moe Matsuki[1], Saki Katagiri[1] and Masaki Ishihara[1]

[1]Fujitsu Limited, 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8588, Japan

Abstract

Behavior change interventions have demonstrated efficacy across numerous domains, yet a significant gap persists between theory and practice. Interventions that are theoretically sound often fail when implemented in real-world applications. This gap is difficult to close because optimizing interventions is hindered by the technical complexity and high cost of development, which slows down iterative improvement cycles. We propose "UMI (Universal Modular Intervention)," a novel framework designed to solve this problem by structurally decoupling the design, implementation, and execution of interventions. We introduce the concept of "Intervention-as-Code," an architecture that dynamically generates application behavior from declarative specifications authored by domain experts. This method allows for modifying intervention strategies and deploying them across different domains without redevelopment, which enhances the speed and flexibility of prototyping. By promoting interdisciplinary collaboration and creating an ecosystem for accumulating and sharing findings, the framework serves as a foundational platform poised to advance the science of behavior change.

Keywords

Behavior Change Support Systems, Intervention-as-Code, Framework

1. Introduction

In recent years, behavior change has demonstrated its effectiveness across a wide range of domains, such as promoting public health [1] and other areas [2]. These techniques, based on insights from psychology and behavioral science, have become widely recognized as important social tools for positively influencing individual decision-making and behavior patterns, thereby improving quality of life (QoL). Particularly with the proliferation of smartphones and wearable devices, Behavior Change Support Systems (BCSS) delivered via digital technology are expected to be a key solution to various challenges, as they can be implemented on a wider scale compared to traditional face-to-face coaching [3, 4, 5]. However, to fully realize their potential, it is crucial to identify effective intervention components and optimize them for individual users and contexts.

Nevertheless, in BCSS research, a "theory-practice gap" is a major challenge, where intervention models that are theoretically sound fail to produce the expected effects in practice [6, 7, 8, 9]. The causes of this gap include implementation issues like technical constraints or developer misinterpretations, and the lack of established methods for applying intervention logic to specific domains [10]. Closing this gap requires rapid improvement cycles, but the enormous time and technical costs of software development create a bottleneck, hindering research progress.

Current BCSS development typically involves building a custom system from scratch for each theory. This meets short-term needs but poses problems with reproducibility and reusability, leading to wasted resources as past work is not utilized. Although systematic implementation methods and frameworks for interventions have been proposed in recent years [11, 12, 13], it remains difficult for other researchers who wish to conduct similar interventions to reproduce and verify the systems of preceding studies, creating a structural flaw that hampers the accumulation of scientific knowledge. This is technically due to the tight coupling of intervention logic with application specifications such as UI and data management, making it difficult to separate and reuse intervention functions.

BCSS 2026: The 14th International Workshop on Behavior Change Support Systems, March 10, 2026, Hakodate, Japan

[*] Corresponding author.

y_tatsuya@fujitsu.com (T. Yamaguchi)
0009-0002-4065-6138 (T. Yamaguchi)

To address these challenges, this study proposes a new framework that separates intervention "design" from "implementation." It provides a mechanism for domain experts to intuitively design interventions by combining various systematized Behavior Change Techniques (BCTs) [14]. This approach allows for rapid testing of different BCT combinations and parameter adjustments with minimal engineering costs. By enabling fast design-verification-evaluation cycles, our framework aims to bridge the theory-practice gap and accelerate the creation of higher-quality evidence in behavior change research.

Research questions. This work addresses the following research questions:

- RQ1: Can intervention design and implementation be structurally decoupled to enable faster iteration in BCSS research?
- RQ2: Can a declarative, executable intermediate representation (Blueprint) express intervention logic based on BCTs?
- RQ3: Can intervention changes (strategy/domain) be localized to Blueprint-level edits without application redevelopment?

Contributions. We contribute: (i) a Designer–Blueprint–Runtime architecture for Intervention-as-Code, (ii) an executable Blueprint representation for BCT-based intervention specifications, and (iii) a preliminary case study demonstrating modifiability and cross-domain reuse.

2. Methods

2.1. Proposed Framework: UMI

In this research, we propose the "UMI (Universal Modular Intervention)" framework as a novel method to enable rapid prototyping of behavior change interventions and the systematic accumulation and sharing of knowledge throughout the community. UMI aims to simultaneously reduce development effort and improve the quality of BCSS by enabling universal and modular intervention definitions that are independent of specific applications or platforms. Unlike traditional monolithic system development, it adopts an approach where intervention elements are treated as independent components that can be freely combined to construct diverse intervention systems. This applies the concepts of component-oriented design from web development and Infrastructure-as-Code from cloud infrastructure to behavior change research, holding the potential to transform the field.

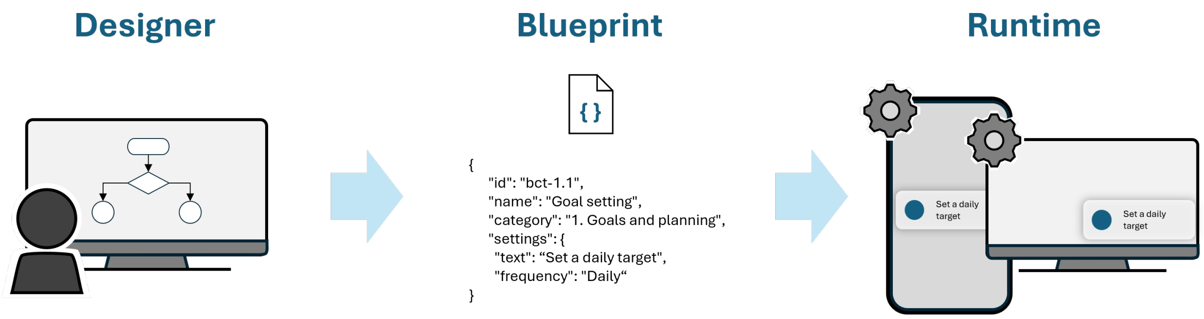
The framework is built on the central concept of "Intervention-as-Code," which involves describing intervention logic not in ambiguous natural language, but as declarative code that can be accurately interpreted by machines. By adopting this approach, an intervention design becomes an executable program. This enables a layered architecture that separates "what" an intervention is from "how" it is implemented, allowing experts and engineers to work independently. This separation makes it possible to incorporate key benefits of modern software engineering into BCSS research, such as version control and automated testing.

2.2. Architectural Components

To realize this concept, the UMI architecture consists of three main components: the "Designer," the "Blueprint," and the "Runtime" (Figure 1).

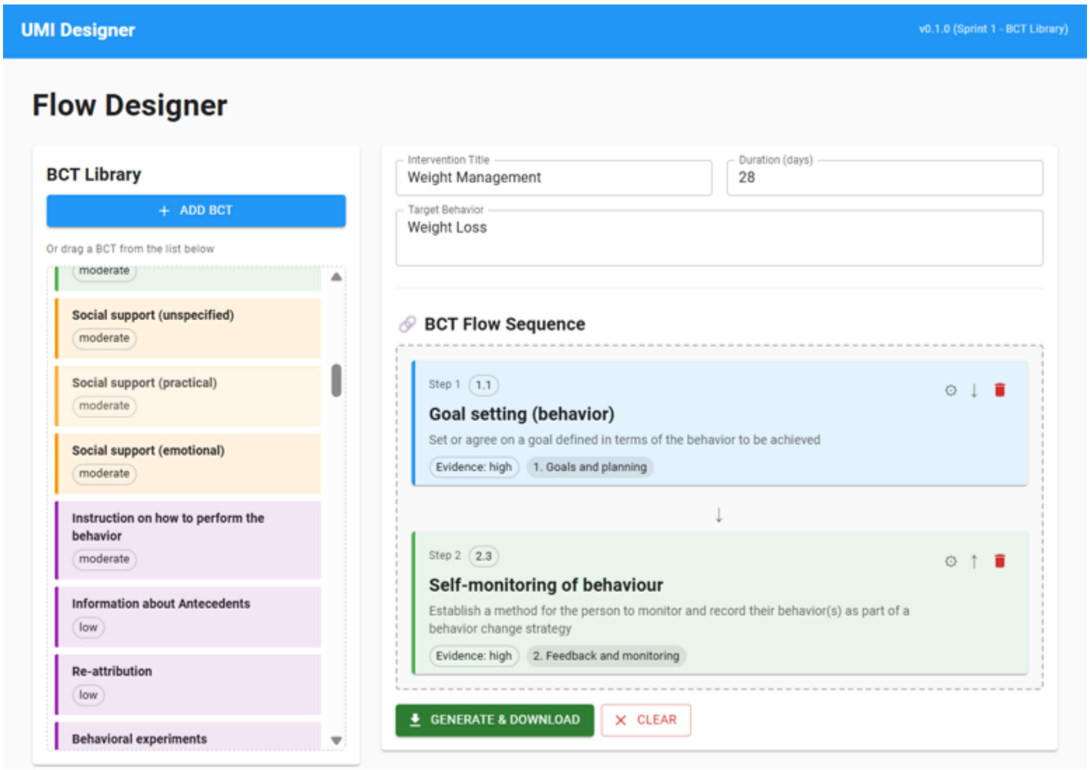
The first layer, "Designer," is a GUI tool that allows domain experts, such as psychologists, to design interventions without writing code. As shown in Figure 2, it enables the visual construction of intervention logic through intuitive drag-and-drop operations from a BCT library. Experts can define the intervention flow and its parameters, and then generate an executable Blueprint file, allowing them to be directly involved in the development process without programming knowledge.

The intervention logic designed in the Designer is output as the second layer, the "Blueprint." This is an intermediate representation file in JSON format and is the core of "Intervention-as-Code." The



```
{
    "id": "bct-1.1",
    "name": "Goal setting",
    "category": "1. Goals and planning",
    "settings": {
      "text": “Set a daily target",
      "frequency": "Daily“
}
```

Figure 1: The overall architecture and workflow of the UMI framework. A domain expert visually designs the intervention logic using the GUI tool "Designer" (left). The design is output as an intermediate representation called "Blueprint" (center), which describes the intervention specifications as code. This code includes the type of BCT and its specific settings. The "Runtime" (right), embedded in the client device, interprets this Blueprint to dynamically generate and control the application’s UI and behavior. This separation enables intervention logic to evolve independently of application implementation.



moderate

Social support (unspecified)
moderate

Social support (practical)
moderate

Social support (emotional)
moderate

Instruction on how to perform the behavior
moderate

Information about Antecedents
low

Re-attribution
low

Behavioral experiments

Intervention Title: Weight Management

Duration (days): 28

Target Behavior: Weight Loss

Step 1 1.1
Goal setting (behavior)
Set or agree on a goal defined in terms of the behavior to be achieved
Evidence: high 1. Goals and planning

Step 2 2.3
Self-monitoring of behaviour
Establish a method for the person to monitor and record their behavior(s) as part of a behavior change strategy
Evidence: high 2. Feedback and monitoring

Figure 2: The user interface of the UMI Designer, a GUI tool for intervention design. Domain experts can browse a library of evidence-based BCTs (left panel) and construct an intervention sequence using drag-and-drop operations (right panel). This allows non-programmers to visually define intervention logic, which is then exported as a machine-readable Blueprint file.

Blueprint describes all specifications, such as the timing of the intervention, the BCTs to be used, and the content to be presented, functioning as an executable specification sheet.

The third layer, "Runtime," is an execution engine that interprets the Blueprint and dynamically generates and controls the UI/UX on the client side. By embedding it as a library into existing applications, it facilitates the addition of behavior change functionalities and the execution of the same intervention logic across different platforms.

2.3. Theory-Based BCT Selection Support

Furthermore, we extended the framework with a feature to support the selection of appropriate BCTs (Behavior Change Techniques) based on theoretical grounds during the initial stage of intervention design. The BCT Taxonomy defines 93 techniques, but determining which technique is effective in which context requires a high level of expertise. UMI addresses this challenge by implementing a recommendation function within the Designer tool used by experts.

The domain expert first inputs the goal the target user wants to achieve (e.g., "to build an exercise habit"). The system interprets this and identifies the necessary psychological processes from the perspective of Mechanisms of Action (MoA) [15] (e.g., "improving self-efficacy"). Next, based on findings from previous research [16], it lists BCTs strongly associated with the identified MoA (e.g., "Goal setting (behavior)") along with their recommendation strength and evidence. This reduces reliance on the expert's experience and supports evidence-based intervention design, thereby increasing the scientific value of the research.

3. Case Study

To qualitatively evaluate the basic practicality, flexibility, and expressive power of the proposed framework, we conducted a case study as a preliminary validation, simulating the development scenario of a typical health behavior change application. One of the authors, acting as a domain expert, used the UMI Designer to create and modify the Blueprints for a mock application. The application, with the UMI Runtime embedded, was then executed on a desktop PC to observe the resulting behavior. The three scenarios simulated for this purpose are outlined in Table 1.

Table 1
Case Study Setup

Case	Scenario	Verification Content
1	Initial Development (Practicality)	Build a weight tracking app from scratch using the Designer and verify its operation.
2	Changed Intervention Strategy (Modifiability)	Modify only the Blueprint to switch the intervention strategy from "goal setting" to "rewards."
3	Cross-Domain Application (Reusability)	While maintaining the core intervention logic of Case 1, adapt its application from "weight management" to "English vocabulary learning."

We quantify editing effort as the number of modified characters between two Blueprint versions, computed after JSON canonicalization with whitespace removed and keys sorted as the sum of inserted and deleted characters in a character-level diff. We report this as a proxy for engineering effort in rapid iteration.

3.1. Results

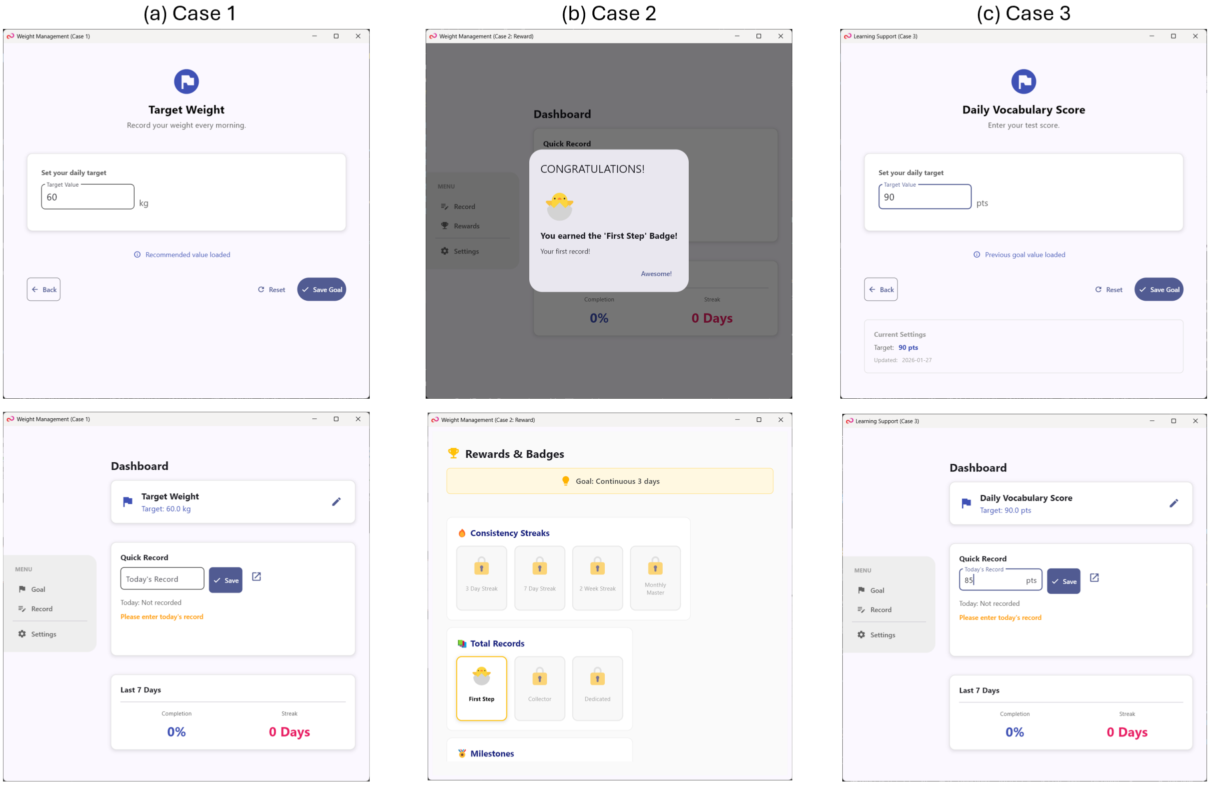
The validation based on the above setup revealed several key points about the UMI framework. First, Case 1 demonstrated the framework's basic practicality. Based solely on the Blueprint definition (Figure 3(a)) generated from the Designer (Figure 2), the intended weight tracking application operated correctly without writing additional application-specific code beyond integrating the Runtime, as shown in the target setting and dashboard screens (Figure 4(a)).

Next, Case 2 confirmed its modifiability. By rewriting only the `bcts` array in the Blueprint (Figure 3(b)), the application's intervention strategy was instantly changed from goal setting to rewards, a change visually confirmed through the new UI elements (Figure 4(b)). This indicates that experimenting with intervention strategies is possible at a substantially reduced engineering effort.

```
// (a) Case 1: Blueprint for Goal Setting
"bcts": [
  {
    "id": "1.1",
    "name": "Goal setting (behavior)",
    "settings_schema": {
      "target_value": 60.0,
      "unit": "kg"
      // ... other settings
    }
  },
  {
    "id": "2.3",
    "name": "Self-monitoring of behavior",
    // ... other settings
  }
]
```

```
// (b) Case 2: Blueprint modified for Reward
"bcts": [
  {
    "id": "2.3",
    "name": "Self-monitoring of behavior",
    // ... other settings
  },
  {
    "id": "10.1",
    "name": "Material incentive (behavior)",
    "settings_schema": {
      "reward_message": "You earned a badge!",
      "rewards_list": [
        {"type": "streak", "value": 3, ...},
        {"type": "streak", "value": 7, ...}
        // ... more reward conditions
      ]
    }
  }
]
```

Figure 3: Example excerpt of the `bcts` array within a Blueprint. (a) on the left shows the intervention definition for Case 1 using goal setting (BCT 1.1). (b) on the right shows the definition for Case 2, modified to use rewards (BCT 10.1). This illustrates that the fundamental change in application behavior, shown in Figure 4, can be achieved by modifying only the BCT definition blocks in the Blueprint.


(a) Case 1

(b) Case 2

(c) Case 3

Target Weight
Record your weight every morning.
Set your daily target
Target Value: 60 kg
Recommended value loaded

Target Weight
Target: 60.0 kg
Today: Not recorded
Please enter today's record

CONGRATULATIONS!
You earned the 'First Step' Badge!
Your first record!

Goal: Continuous 3 days

Daily Vocabulary Score
Enter your test score.
Set your daily target
Target Value: 90 pts
Previous goal value loaded
Current Settings
Target: 90 pts
Updated: 2026-01-27

Daily Vocabulary Score
Target: 90.0 pts
Today's Record: 85 pts
Today: Not recorded
Please enter today's record

Figure 4: Application screenshots dynamically generated by the UMI Runtime, demonstrating the results of the three case study scenarios. (a) Case 1: The initial weight management app, showing the target setting and dashboard screens generated from the Blueprint. (b) Case 2: The app's behavior modified for a reward-based strategy, showing a badge acquisition pop-up and a rewards dashboard. (c) Case 3: The core intervention logic reused for an English vocabulary learning app. These results visually illustrate the framework's practicality, modifiability, and reusability.

Finally, Case 3 proved its high reusability. Using the Blueprint from Case 1 as a template, the same intervention logic was easily applied to a different domain by simply changing domain-specific parameters (e.g., from "kg" to "pts"), successfully generating an English vocabulary learning application

(Figure 4(c)).

As shown in Table 2, switching the intervention strategy required 1998 modified characters in the Blueprint, whereas cross-domain adaptation required 530, indicating that both changes were localized to Blueprint-level edits without application redevelopment.

Table 2
Blueprint editing effort measured as modified characters (insertions + deletions) after JSON canonicalization.

Transition	**Change**	Artifact(s) changed	Blueprint size (chars)	Modified chars (ins+del)
1→2	Switch intervention strategy (goal setting → reward)	1 (Blueprint)	1049→2061	1998
1→3	Cross-domain adaptation (weight → vocabulary learning)	1 (Blueprint)	1049→1015	530

4. Discussion

This case study suggests that the UMI framework has the potential to address the deep-rooted problems of slow iteration and limited flexibility in BCSS development. Importantly, the evaluation focuses on development-level outcomes rather than behavioral outcomes. In particular, both switching the intervention strategy and cross-domain adaptation were achieved through localized edits to the Blueprint (Table 2) without modifying the host application code beyond integrating the Runtime. No application-layer source code beyond the initial Runtime integration was modified in either transition. While the absolute time savings require validation in real development settings, these results indicate that UMI can reduce engineering burden during early-stage prototyping, which traditionally took weeks to months [13], and iterative exploration.

The core contribution of this method lies in providing an implementation foundation that serves as a "common language" among diverse experts such as psychologists, healthcare professionals, and engineers. As human-readable code, the Blueprint facilitates smooth collaboration across disciplinary boundaries. Furthermore, by managing it with version control systems like Git, the history of "who, when, and why an intervention was changed" can be strictly tracked, technically guaranteeing the reproducibility of research. The complete reproduction of an intervention, which was difficult with only descriptions in papers, becomes possible by sharing the Blueprint file.

UMI does not assume that BCTs are independent or universally composable. Instead, it encapsulates each BCT into a distinct, parameterized unit. This architecture enables researchers to implement and empirically test different combinations and sequences with minimal redevelopment cost. Consequently, potential interaction effects or conflicts between BCTs are not an architectural limitation, but rather a central part of the experimental design space that UMI empowers researchers to explore.

Positioning with prior frameworks. UMI is complementary to prior frameworks as it provides the technical implementation layer they do not specify. While influential models offer conceptual guidance and process methodologies, UMI provides the architectural substrate to put them into practice efficiently. For instance, the Behavioral Intervention Technology (BIT) model [12] offers a conceptual language (why/what/how) to deconstruct interventions, and the IDEAS framework [13] provides a procedural workflow for their design and evaluation. These frameworks define what should be done, but not how to implement it in a rapidly modifiable way. UMI, in contrast, focuses on the how of implementation. It operationalizes the intervention logic—the what defined by models like BIT—into an executable, version-controllable Blueprint file. This allows the intervention itself to be modified and redeployed through simple Blueprint edits, enabling the rapid iteration central to IDEAS without costly application redevelopment. Thus, UMI acts as the engine that makes these established theoretical and procedural goals technically feasible.

4.1. Limitations and Future Work

However, this study has several limitations. The current evaluation is limited to a qualitative analysis based on a restricted scenario case study. Further verification is needed regarding its effectiveness in large-scale field experiments with actual users. Additionally, whether the usability of the Designer and the expressive power of the Blueprint can handle all types of complex interventions is another evaluation issue. Real-world behavior change is highly context-dependent, and situations requiring subtle nuances or dynamic adaptations that cannot be captured by the current model are conceivable.

In addition, the current validation did not involve external domain experts or end users; therefore, usability, learning curve, and perceived development effort reduction should be evaluated in independent studies.

Future work includes conducting field studies in real-world environments over a long-term period to demonstrate the framework's effectiveness through quantitative evaluation of intervention persistence and user engagement. Concurrently, it is necessary to expand the types of BCTs currently supported and to develop a module library that can handle more diverse and advanced intervention logic.

5. Conclusion

In this study, to bridge the gap between theory and practice in BCSS development, we proposed the UMI framework, centered on "Intervention-as-Code," and demonstrated its effectiveness through a case study. By separating the design, implementation, and execution of interventions, this framework provides a common foundation that promotes rapid prototyping, improves research reproducibility, and facilitates collaboration between experts and engineers, showing a path to solving the cost and rigidity problems of conventional development. We anticipate that as this framework is widely used and knowledge is accumulated, more effective BCSS may become more widely adopted in society.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini 2.5 Pro in order to: Text Translation, Grammar and spelling check. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] American Diabetes Association Professional Practice Committee, 5. facilitating behavior change and well-being to improve health outcomes: Standards of medical care in diabetes—2022, Diabetes Care 45 (2022) S60–S82. URL: https://diabetesjournals.org/care/article/45/Supplement_1/S60/138923/5-Facilitating-Behavior-Change-and-Well-being-to. doi:10.2337/dc22-S005.

[2] B. Verplanken, S. Orbell, Attitudes, habits, and behavior change, Annual Review of Psychology 73 (2022) 327–352. URL: https://www.annualreviews.org/doi/10.1146/annurev-psych-020821-011744. doi:10.1146/annurev-psych-020821-011744.

[3] J. Cane, D. O'Connor, S. Michie, Validation of the theoretical domains framework for use in behaviour change and implementation research, Implementation Science 7 (2012) 37. URL: https://implementationscience.biomedcentral.com/articles/10.1186/1748-5908-7-37. doi:10.1186/1748-5908-7-37.

[4] A. M. Roffarello, L. D. Russis, Achieving digital wellbeing through digital self-control tools: A systematic review and meta-analysis, ACM Transactions on Computer-Human Interaction 30 (2023) 1–66. URL: https://dl.acm.org/doi/10.1145/3571810. doi:10.1145/3571810.

[5] A. O. Ekpezu, I. Wiafe, H. Oinas-Kukkonen, Enhancing perceived health competence: The impact of persuasive social support features in health and fitness apps, International Journal of

Human–Computer Interaction 40 (2024) 8076–8090. URL: https://www.tandfonline.com/doi/full/10.1080/10447318.2023.2277493. doi:10.1080/10447318.2023.2277493.

[6] S. Michie, M. Johnston, J. Francis, W. Hardeman, M. Eccles, From theory to intervention: Mapping theoretically derived behavioural determinants to behaviour change techniques, Applied Psychology 57 (2008) 660–680. URL: https://iaap-journals.onlinelibrary.wiley.com/doi/10.1111/j.1464-0597.2008.00341.x. doi:10.1111/j.1464-0597.2008.00341.x.

[7] L. Colusso, Behavior change theory and design, in: Proceedings of the 2018 ACM Conference Companion Publication on Designing Interactive Systems, ACM, 2018, pp. 343–346. URL: https://dl.acm.org/doi/10.1145/3197391.3205378. doi:10.1145/3197391.3205378.

[8] J. Butler, H. Asbridge, H. Stringer, Applying behaviour change theory to speech and language therapy intervention for inducible laryngeal obstruction, International Journal of Language & Communication Disorders 58 (2023) 1539–1550. URL: https://onlinelibrary.wiley.com/doi/10.1111/1460-6984.12880. doi:10.1111/1460-6984.12880.

[9] A. Ananthakrishnan, C. Cong, V. Riccalton, E. Meinert, Progressing implementation of behavior change frameworks for digital health interventions: challenges and ways forward, Translational Behavioral Medicine 15 (2025). URL: https://academic.oup.com/tbm/article/doi/10.1093/tbm/ibaf069/8378415. doi:10.1093/tbm/ibaf069.

[10] G. Marcu, S. J. Ondersma, A. N. Spiller, B. M. Broderick, R. Kadri, L. R. Buis, Barriers and considerations in the design and implementation of digital behavioral interventions: Qualitative analysis, Journal of Medical Internet Research 24 (2022) e34301. URL: https://www.jmir.org/2022/3/e34301. doi:10.2196/34301.

[11] E. E. Y. F. Agyei, M. Kekkonen, H. Oinas-Kukkonen, Evaluating the persuasive potential from software design specifications, in: 19th International Conference on Persuasive Technology, 2024, pp. 15–25. URL: https://link.springer.com/10.1007/978-3-031-58226-4_2. doi:10.1007/978-3-031-58226-4_2.

[12] D. C. Mohr, S. M. Schueller, E. Montague, M. N. Burns, P. Rashidi, The behavioral intervention technology model: An integrated conceptual and technological framework for ehealth and mhealth interventions, Journal of Medical Internet Research 16 (2014) e146. URL: http://www.jmir.org/2014/6/e146/. doi:10.2196/jmir.3077.

[13] S. A. Mummah, T. N. Robinson, A. C. King, C. D. Gardner, S. Sutton, Ideas (integrate, design, assess, and share): A framework and toolkit of strategies for the development of more effective digital interventions to change health behavior, Journal of Medical Internet Research 18 (2016) e317. URL: http://www.jmir.org/2016/12/e317/. doi:10.2196/jmir.5927.

[14] S. Michie, M. Richardson, M. Johnston, C. Abraham, J. Francis, W. Hardeman, M. P. Eccles, J. Cane, C. E. Wood, The behavior change technique taxonomy (v1) of 93 hierarchically clustered techniques: Building an international consensus for the reporting of behavior change interventions, Annals of Behavioral Medicine 46 (2013) 81–95. URL: https://academic.oup.com/abm/article/46/1/81/4563254. doi:10.1007/s12160-013-9486-6.

[15] R. N. Carey, L. E. Connell, M. Johnston, A. J. Rothman, M. de Bruin, M. P. Kelly, S. Michie, Behavior change techniques and their mechanisms of action: A synthesis of links described in published intervention literature, Annals of Behavioral Medicine (2018). URL: https://academic.oup.com/abm/advance-article/doi/10.1093/abm/kay078/5126198. doi:10.1093/abm/kay078.

[16] M. Johnston, R. N. Carey, L. E. C. Bohlen, D. W. Johnston, A. J. Rothman, M. de Bruin, M. P. Kelly, H. Groarke, S. Michie, Development of an online tool for linking behavior change techniques and mechanisms of action based on triangulation of findings from literature synthesis and expert consensus, Translational Behavioral Medicine 11 (2021) 1049–1065. doi:10.1093/tbm/ibaa050.