Digital Health and Informatics Innovations for Sustainable Health Care Systems
J. Mantas et al. (Eds.)

doi:10.3233/SHTI240350

Identifying Reusable Core Assets of Digital Health Apps

Richard MAY[a], Robin GLAUSER[b] and Kerstin DENECKE[b,1]
[a] *Harz University of Applied Sciences, Wernigerode, Germany*
[b] *Bern University of Applied Sciences, Institute Patient-centered Digital Health, Biel/Bienne, Switzerland*
ORCiD ID: Richard May https://orcid.org/0000-0001-7186-404X
Robin Glauser https://orcid.org/0000-0003-0229-4375
Kerstin Denecke https://orcid.org/0000-0001-6691-396X

Abstract. Despite their variability, Digital Health Apps (DHAs) typically share functionalities (i.e. core assets) and can thus be considered as a family of similar products with unique features adapted to specific use cases. *Objective:* We aim to identify and model reusable core assets to facilitate the development of a number of similar, but adapted DHAs in the context of an initial product line engineering approach. *Methods:* To identify core assets, we apply a systematic analysis of six exemplary state-of-the-art DHAs. In an iterative process, they were modeled in a feature model. *Results*: We identified 14 core assets of DHAs out of which six are mandatory (i.e. required by each DHA) and eight are optional core assets (i.e. required by most DHAs, depending on the app complexity). *Conclusions:* We found that DHAs share common functionalities that could contribute to a more efficient development of the DHA, especially in terms of time and cost savings.

Keywords. Digital health apps, product line engineering, feature modeling

1. Introduction

The digital health landscape is rapidly evolving, driven by technological advances and an increasing demand for accessible healthcare services. In this context, Digital Health Apps (DHAs) have emerged as a critical component, aiming to provide users with reliable, on-demand health management tools, e.g. suicide prevention [1] or medication management [2]. DHAs are becoming increasingly configurable, including a variety of features (i.e. user-visible functionalities [3]) adapted to stakeholder needs. Despite their variability, DHAs often share *core assets* (i.e. similar functionalities) and thus can be considered as a family of quite similar products but with unique features adapted to specific use cases [4]. Unfortunately, the increasing configurability of these systems leads to a growing system complexity [5], which makes their development quite challenging, as systems typically need to be completely redesigned for new use cases. Therefore, there is a great need for strategies to handle configurability and reuse components, in particular to save time and costs in development, and to ensure high scalability despite the richness of features and DHA variants [6].

[1] Corresponding Author: Kerstin Denecke, Bern University of Applied Sciences, Institute Patient-centered Digital Health, Quellgasse 21, Biel, Switzerland; E-mail: kerstin.denecke@bfh.ch.

Product Line Engineering (PLE) is a strategy that helps organizations to efficiently develop multi-variant applications using an integrated platform. It offers benefits such as improved product quality and faster, more cost-effective delivery [7,8]. PLE relies on feature modeling to manage variability, enabling the derivation and verification of application variants and their configurations [9]. It comprises three main phases: domain engineering for platform design and requirements [3]; security engineering for designing security measures and certifying configurations [10]; and application engineering for implementing variants according to customer requirements. PLE also includes a framework of problem space (domain abstractions), solution space (implementation of abstractions) and a mapping between them to guide configuration [3].

This paper presents an initial PLE approach tailored to the development of similar, but customized DHAs. PLE is considered ideal for the digital health sector, as it enables the creation of diverse health applications using common core assets, allowing customization for specific health domains or stakeholder needs. Examples include Gomes et al. [11] and Losavio et al. [12] who designed product lines for health information systems to create system architectures, and Hitesh et al. who optimized feature selection for a health application product line [13]. While there is existing research on PLE in healthcare, it often focuses on broader contexts, may be outdated, or overlooks recent trends such as mobile apps or conversational technologies (e.g. large language models) that are relevant to DHAs [6].

The goal of our paper is to identify and model core assets of DHAs, focusing specifically on the problem space during the domain engineering phase of PLE. We believe that the concept of PLE cannot only streamline the development process of DHAs, but also improve their quality, scalability and sustainability.

2. Methods

We have two subordinate goals in this paper: (1) the identification of core assets to determine the product line scope and (2) the modeling of core assets in terms of a feature model. To identify DHA core assets, we rely on a systematic analysis of six exemplary state-of-the-art DHAs that share similar functionalities but are adapted to their specific use cases. Namely we investigate the DHAs *SERO [1]*, *SFK MindCare*, *Approches, eMMA [2], CLAIRE [14]* and *Ana [15]*.

- SERO: Suicide prevention app with visual assessment tool and personalized safety plan
- SFK Mindcare: App to train self-awareness in people with an addiction disorder; offers diary for craving and urges, tips and exercises for self-regulation (link: https://www.bfh.ch/en/research/research-projects/2019-763-030-755/)
- Approches: App that allows informal caregiver to access the support services (e.g. meal delivery, home transport etc.) available in a specific region (link: http://www.approches.ch)
- eMMA: App that supports medication management with integrated chatbot
- Claire: App that provides virtual reality patient education with voice interaction
- Ana: App to collect the music biography of a person based on a chatbot

To extract data, the first two authors summarized the DHA features as free-text and used open-card-sorting to synthesize recurring categories. Technical variability features from previous research were used as a rough guideline, which, however, refer exclusively to conversational agents [6]. The second author then created a feature model based on the identified core assets. In three rounds involving all authors, the selected core assets and the feature model were discussed and continuously revised. Disagreements were discussed until a consensus between all authors was reached.

3. Results

Overall, we identified 14 core assets of DHAs (cf. Figure 1). These include six mandatory (i.e. required by each DHA) and eight optional core assets (i.e. required by most DHA, depending on the app complexity). In the following paragraphs, we have indicated in brackets the number of apps that contain a particular core asset.

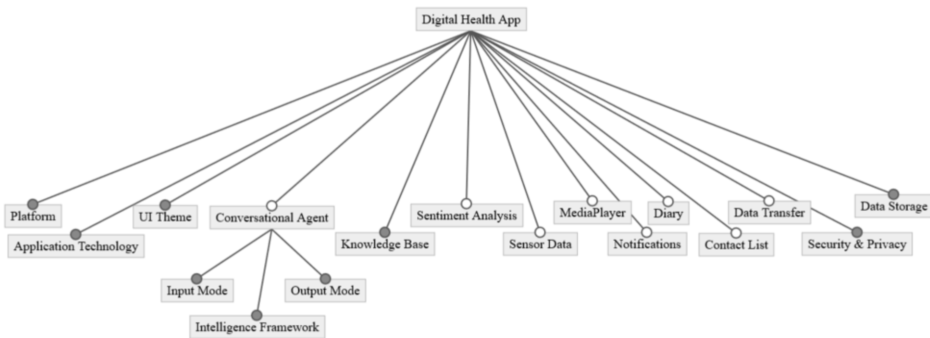

Figure 1. DHA feature model, including mandatory (gray circle) and optional (white circle) core assets.

Mandatory core assets. DHAs are deployed on a *platform* (6), which typically relates to mobile devices (e.g. Android) or web. Moreover, DHAs are usually based on a specific *application technology* (6), such as virtual reality, vocal or normal. Every DHA is built on a specific design, which we refer to as a *user interface theme* (6), e.g. buttons, font or spacing. The theme has a direct impact on the user experience and a great potential of reusing designs that are easy to understand and efficient to use. In addition, DHAs usually have a *knowledge base* (6) consisting of use case-specific information as well as general information. Every DHA requires data that must be stored accordingly. In this context, *data storage* (6) becomes relevant, which is either realized internally or externally (e.g. outsourcing to third parties). For similar use cases, reusing storage environments (e.g. cloud services) and storage media (e.g. SQL/NoSQL database) is a suitable option. As sensitive data is handled in the digital health domain, data *security and privacy* (3) mechanisms are essential (e.g. authentication).

Optional core assets. A current trend in DHA development is the implementation of *conversational agents* (3), which can perform various tasks in the context of information transfer. Conversational agents rely on three mandatory core assets, namely *input* and *output mode* (e.g. text) as well as *intelligence framework* (rule-based vs. machine learning). Closely related to general DHA core assets requiring user inputs is the app's appropriate response to a user's emotions in the context of *sentiment analysis* (0), which, however, is currently not common in most DHA due to several challenges (e.g. reliable emotion interpretation). *Sensor data* (1) refer to features related to internal or external sensors (e.g., Fitbit) that can be connected, e.g. via built-in APIs, Bluetooth, Wi-Fi or NFC. To deliver media content to users, many applications provide *media players* (1) which can be included as a reusable feature in related products. Moreover, there are *notification services* (1) for user engagement, but also for information provision (e.g. medication reminders). Additional core assets include *diary* features (3), and *contact lists* (3), including contact information on health professionals. DHAs occasionally require data to be transferred, highlighting the importance of secure *data transfer* (5) mechanisms. For example, data transfer to external devices or software requires interoperability and appropriate data transfer capabilities, which can be effectively implemented using HL7 FHIR resources.

4. Discussion

We emphasize that reusing core assets offers significant advantages, including reducing development time and costs, realizing user experience across similar applications, or simplified app maintenance. However, there are various challenges that need to be addressed. Currently, our feature model does not consider all potential dependencies, in particular those that are related to the compatibility between features. In the worst case, unwanted feature interactions or misconfigurations may lead to security vulnerabilities or app failure [16]. Such issues are critical in the context of safety-critical apps (e.g. medication management) [17]. So, we need strategies enabling the automated derivation of verified configurations (i.e. DHA variants) and considering evolutionary changes during the development lifecycle [18]. Another issue arises regarding the increasing number of core assets due to technological developments. For instance, recent developments in the field of large language models have not only added numerous new dependencies (i.e. technological and legal), but also various new (reusable) features. Thus, DHAs are becoming increasingly complex, leading to a feature explosion [19] and thus DHA variants. So, analyses are needed to reduce configuration options to a permissible and maintainable number. In addition, technological progress, system evolution but also legal regulations (e.g. GDPR, ISO/IEC 27002) lead to challenges in the context of feature model timeliness. Accordingly, feature models themselves need to be maintained, which should be realized through (semi-)automated processes (e.g. product-based verification) [20]. Overall, the question arises to what extent the introduction of a holistic PLE process for DHAs results in the benefits outlined before and overcomes the challenges. Accordingly, our main goal for further research is to address the DHA security engineering and application engineering phases in the mapping and solution space, i.e. addressing the automated generation of new DHAs based on core assets.

5. Conclusions

In this paper, we presented an initial PLE approach for DHAs. Specifically, six apps were investigated in detail to identify their core assets as common, reusable features and to create a feature model. We believe that the concept of PLE could be a key strategy in developing DHAs more efficiently, in particular in terms of saving time and costs. It allows DHA development to adapt quickly to new digital health trends, regulatory changes and technological advances. Further research includes investigating the link between modeling and the generation of architectural concepts and implementations, and analyzing to what extent PLE makes the development of DHAs more efficient compared to traditional development approaches.

References

[1] Meier L, Gurtner C, Nuessli S, Miletic M, Bürkle T, Durrer M. SERO - A New Mobile App for Suicide Prevention. Stud Health Technol Inform 2022 May; 292: 3–8.
[2] Tschanz M, Dorner TL, Holm J, Denecke K. Using eMMA to Manage Medication. Computer 2017 Aug.; 51(8):18–25, doi: 10.1109/MC.2018.3191254.
[3] Apel S, Batory D, Kästner C, Saake G. Feature-Oriented Software Product Lines: Concepts and Implementation. Berlin, Heidelberg: Springer, 2013.
[4] Pohl K, Böckle G, and Van Der Linden F. Software Product Line Engineering. Berlin, Heidelberg: Springer, 2005, doi: 10.1007/3-540-28901-1.
[5] Jamshidi P, et al. Transfer Learning for Improving Model Predictions in Highly Configurable Software. 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Buenos Aires, Argentina: IEEE, May 2017. p. 31–41.
[6] May R, Denecke K. Conversational Agents in Healthcare: A Variability Perspective. Proceedings of the 18th International Working Conference on Variability Modelling of Software-Intensive Systems, Bern Switzerland; ACM, Feb. 2024. p. 123–128, doi: 10.1145/3634713.3634717.
[7] Krüger J, Berger T. An empirical analysis of the costs of clone- and platform-oriented software reuse. Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Virtual Event USA: ACM, Nov. 2020. p. 432–444. doi: 10.1145/3368089.3409684.
[8] van der Linden F, Schmid K, and Rommes E. Software product lines in action: the best industrial practice in product line engineering. Berlin, New York: Springer, 2007.
[9] Nešić D, Krüger J, Stănciulescu S, and Berger T. Principles of feature modeling. 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn Estonia: ACM, Aug. 2019. p. 62–73.
[10] May R, Biermann C, Kenner A, Krüger J, and Leich T. A product-line-engineering framework for secure enterprise-resource-planning systems. Int. Conf. on ENTERprise Information Systems, 2023. p. 1–8.
[11] Gomes ATA, Ziviani A, Correa BSPM, Teixeira IM, and Moreira VM. 'SPLiCE: a software product line for healthcare. Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, Miami Florida USA: ACM, Jan. 2012. p. 721–726. doi: 10.1145/2110363.2110447.
[12] Losavio F, Ordaz O. Reference Architecture Representation by an Ontology for Healthcare Information Systems Software Product Line. 4to Simposio SCTC, 2016. p. 20–32.
[13] Chhikara HR, Kumari AC. Feature Selection Optimization of HealthCare Software Product Line using BBO. Procedia Computer Science, 2020; 167:1696–1704.
[14] May R, Denecke K. Extending Patient Education with CLAIRE: An Interactive Virtual Reality and Voice User Interface Application. Alario-Hoyos C et al, editors. Addressing Global Challenges and Quality Education. Lecture Notes in Computer Science; Springer International Publishing, Cham; 2020;12315: 482–486. doi: 10.1007/978-3-030-57717-9_49.
[15] Denecke K, Hochreutener S, Pöpel A, and May R. Self-Anamnesis with a Conversational User Interface: Concept and Usability Study. Methods Inf Med. 2018 Nov.; 57(05/06):243–252
[16] May R, et al. Vulnerably (Mis) Configured? Exploring 10 Years of Developers' Q&As on Stack Overflow. 18th International Working Conf. on Variability Modelling of Software-Intensive Systems, 2024. p.112-122.
[17] May R, et al. A systematic mapping study on security in configurable safety-critical systems based on product-line concepts. ICSOFT. SciTePress. 2023:217-224.
[18] Biffl S, Mordinyi R, Moser T. Automated Derivation of Configurations for the Integration of Software (+) Engineering Environments. InACoTA 2010: 6-13.
[19] Zhu G, et al. Evolutionary automated feature engineering. InPacific Rim International Conference on Artificial Intelligence 2022 Nov 4. Cham: Springer Nature Switzerland, 2022. p. 574-586.
[20] de Mello RM, et al. Verification of Software Product Line Artefacts: A Checklist to Support Feature Model Inspections. J. Univers. Comput. Sci. 2014; *20*(5): 720-745.