End-User Programming of Mobile Services: Empowering Domain Experts to Implement Mobile Data Collection Applications

Johannes Schobel[1], Rüdiger Pryss[1], Marc Schickler[1], Martina Ruf-Leuschner[2], Thomas Elbert[2], Manfred Reichert[1]
[1]*Institute of Databases and Information Systems, Ulm University, Germany*
[2]*Department of Psychology, University of Konstanz, Germany*
[1]{*johannes.schobel, ruediger.pryss, marc.schickler, manfred.reichert*}@*uni-ulm.de*
[2]{*martina.ruf, thomas.elbert*}@*uni-konstanz.de*

***Abstract*—The widespread use of smart mobile devices (e.g., in clinical trials or online surveys) offers promising perspectives with respect to the controlled collection of high-quality data. The design, implementation and deployment of such mobile data collection applications, however, is challenging in several respects. First, various mobile operating systems need to be supported, taking the short release cycles of vendors into account as well. Second, domain-specific requirements need to be flexibly aligned with mobile application development. Third, usability styleguides need to be obeyed. Altogether, this turns both programming and maintaining mobile applications into a costly, time-consuming, and error-prone endeavor. To remedy these drawbacks, a model-driven framework empowering domain experts to implement robust mobile data collection applications in an intuitive way was realized. The design of this end-user programming framework is based on experiences gathered in real-life mobile data collection projects. Facets of various stakeholders involved in such projects are discussed and an overall architecture as well as its components are presented. In particular, it is shown how the framework enables domain experts (i.e., end users) to flexibly implement mobile data collection applications on their own. Overall, the framework allows for the effective support of mobile services in a multitude of application domains.**

***Keywords*-Mobile Data Collection, Process-aware Information System, Process Flexibility.**

I. Introduction

In the light of trends like big data and cloud computing, mobile technology has become a salient factor for projects that have hitherto collected data on a paper basis. The latter, in turn, varies from simple to-do lists up to complex instruments (e.g., medical questionnaires), which are required in numerous application domains (e.g., healthcare and psychology). To provide a generic approach for mapping paper-based instruments to mobile applications, however, profound insights into real-world scenarios are indispensable. Ideally, these insights are gathered during long-running, large-scale scenarios. The development of the framework presented in this paper was based on the experiences gained in the context of 8 mobile applications scenarios dealing with data collection in the large scale (cf. Table I). In these projects, domain experts were provided with specifically tailored mobile applications instead of traditional paper-based collection instruments. Note that all 8 projects revealed that, when using mobile applications, data can be collected more conveniently compared to paper-based approaches. Furthermore, in most projects a large amount of data was collected in a rather short time period. For example, the *TrackYourTinnitus* project gathered more than 200,000 data items from 20,000 processed questionnaire instances within one year (cf. Table I) based on crowd sensing technologies [1]. Compared to traditional ways of paper-based data collection, data quality could be significantly increased. Accordingly, the information value the collected data has for domain experts could be enhanced as well.

Data Collection Scenario	Country	CN	Releases	Instances
Tinnitus Research [1]	World-Wide	○	3	$\geq$ 20,000
Risk Factors during Pregnancy [2]	Germany	○	5	$\geq$ 1,000
Risk Factors after Pregnancy	Germany	○	1	$\geq$ 100
PTSD in War Regions [3]	Burundi	●	5	$\geq$ 2,200
PTSD in War Regions [4]	Uganda	○	1	$\geq$ 200
Adverse Childhood Experiences [5]	Germany	●	3	$\geq$ 150
Learning Deficits among Medical Students	Germany	●	3	$\geq$ 200
Supporting Parents after Accidents of Children	Switzerland, Germany	○	5	$\geq$ 2,500
Sum Σ			24	$\geq$ 26,350

CN = Complex Navigation; PTSD = Posttraumatic Stress Disorder

Table I
Realized Mobile Data Collection Applications

As a lesson learned, domain experts were not completely satisfied with the functionality provided by these hard-coded prototypes. In particular, they craved for functions beyond the capabilities of paper-based instruments. For example, they demanded complex navigation operations guiding untrained staff through the *process of data collection* (e.g., automatically skipping questions depending on already given answers (cf. Table I, column *CN*). In turn, respective change requests required new releases of the already existing mobile application (cf. Table I, column *Releases*). Furthermore, additional releases became necessary due to other reasons. For example, multilingual user interfaces need to be provided. In this context, the Burundi mobile application project (cf. Table I) was most challenging: In this project, the work of domain experts could be dramatically eased by providing

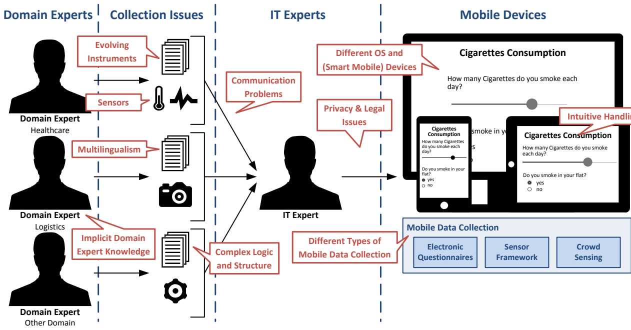

Figure 1. Fundamental Facets of Mobile Data Collection

complex navigation operations to them when filling in the questionnaires. Altogether, five major releases became necessary during the project. Note that the maintenance of the complex navigation logic for the five releases resulted in a cost explosion with respect to the project budget.

The discussed requirements (e.g., frequent releases, multilingualism, need for flexible navigation) need to be considered in combination with each other, requiring a proper categorization. Fig. 1 summarizes key facets of the various projects. These facets, in turn, provide the basis for the development of a framework empowering domain experts to flexibly realize advanced mobile data collection applications themselves. In particular, the framework includes end-user programming techniques for this purpose.

The *first facet* concerns the proper involvement of *domain experts*. Note that the projects from Table I revealed that the respective application domain poses specific requirements. While in certain projects, multilingualism was an issue, in others the support of complex navigation operations within the data collection application was demanded.

The *second facet* deals with *technical issues related to data collection*. In several projects, sensors and wearables needed to be integrated. Recorded sensor data, in turn, enabled experts to interpret and evaluate the gathered data more properly. For example, measuring the heart rate of a subject during the processing of a questionnaire results in additional valuable and accurate information when conducting clinical trials. As another requirement, the data collected with smart mobile devices needed to be transferred to appropriate data analysis tools.

The *third facet* refers to the proper involvement of *IT experts*. In this context, two requirements were identified: First, the technical communication between smart mobile devices and external services is essential, taking privacy issues into account as well. Regarding healthcare projects, for example, collected data frequently comprises sensitive patient data and, therefore, needs to be encrypted [6]. Second, all projects needed to cope with the *business IT alignment gap*; i.e., the requirements of the domain experts needed to be properly mapped to the mobile applications. Initially, the semantics of the paper-based instruments was not evident for IT experts. Consequently, profound domain knowledge is required to correctly transfer a paper-based instrument and its logic to a mobile data collection application. To bridge this gap, domain experts should be empowered to realize mobile data collection applications on their own.

The final facet deals with *smart mobile device services*. Three particular requirements need to be considered for them. First, the functionality of a data collection instrument must be provided on various mobile operating systems. Second, major vendor release cycles of mobile operating systems, which are rather frequent, need to be properly handled. Third, adaptations with respect to different screen sizes (e.g., tablet or smartphone) are crucial.

This paper introduces a generic framework based on the introduced facets empowering domain experts to realize sophisticated mobile data collection applications. Section II presents fundamentals for this paper. In Section III, the overall architecture of the developed framework is presented, while Section IV introduces a powerful component for configuring mobile data collection instruments. Section V presents preliminary results of a study applying this configurator component in practice. Section VI discusses related work and Section VII presents conclusion and future work.

II. Fundamentals

Fig. 2 introduces the mobile data collection lifecycle, which consists of 5 phases. Note that in 3 phases end-user programming techniques are applied. They, in turn, provide the basis for empowering domain experts to realize sophisticated mobile data collection applications themselves.

In the *Design & Modeling* phase, mobile data collection instruments with complex navigation logic are created by end-users. The *Deployment* phase, in turn, allows for the robust deployment of the created instruments to smart mobile devices. In the *Enactment & Execution* phase, multiple instances of the realized data collection instruments may be created and executed on the smart mobile devices. During

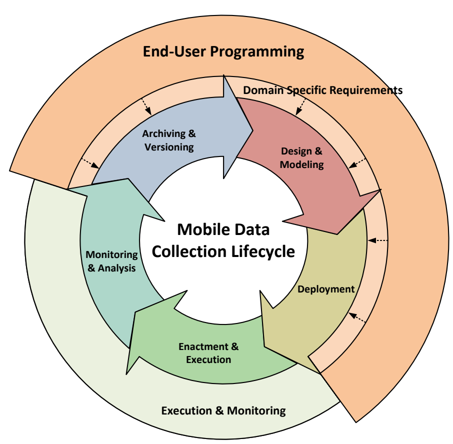

Figure 2. Mobile Data Collection Lifecycle

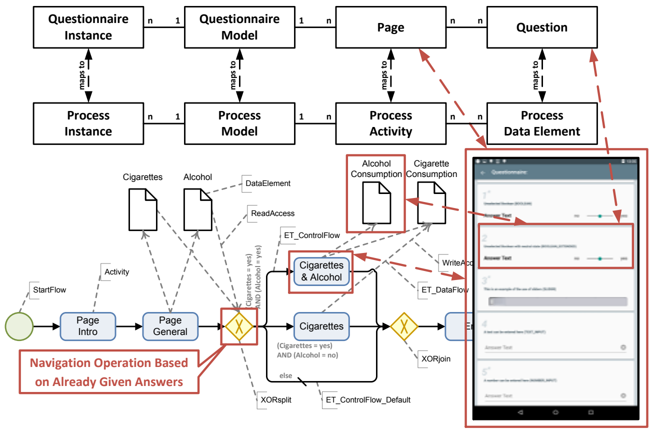

Figure 3. Relation Between a Questionnaire and a Process Model

the *Monitoring & Analysis* phase, collected data is analyzed in real-time on the smart mobile device and the backend system respectively. Finally, the *Archiving & Versioning* phase provides advanced techniques for managing release cycles of mobile application services as well as for versioning them.

The presented work focuses on the *Design & Modeling* and *Deployment* phases. In particular, a generic model-driven configurator approach for creating mobile data collection applications for various domains is presented.

In order to transfer paper-based instruments to digital ones, first of all, a mental model was defined (cf. Fig. 3). Thereby, the logic of an instrument is described in terms of an executable process model that can be enacted by a lightweight process engine running on smart mobile devices of different operating systems. This model-driven approach, in turn, separates process logic from application code [7]. A process model, in turn, acts as schema for executing process instances (e.g., questionnaire instances). The process model itself consists of process steps (i.e., activities) as well as the control and data flow between them. Furthermore, gateways (e.g., XORsplit and ANDsplit) are provided for describing control flow structures.

Using this mental model, both the logic and content of a paper-based instrument can be mapped to a process model. More precisely, *pages* of an instrument correspond to *process activities* and the flow between these activities matches the navigation logic of the instrument. *Questions*, in turn, are directly mapped to *process data elements*, which, in turn, are connected to activities. The latter may write data elements to store the answers to specific questions. More details about mapping an instrument to a process model can be obtained from [8]. The structure capturing all required information relies on the ADEPT2 [9] process model, but can also be adapted to other meta-models (e.g., WS-BPEL [10]).

III. Architecture

This section describes the generic architecture of the realized framework for managing mobile data collection applications (cf. Fig. 4). This architecture, in turn, relies on a process-driven approach. The latter constitutes the basis for coping with fundamental requirements and technical challenges (cf. Section I). In the following, it is discussed how process management technology drives this architecture:

1) *Create Collection Instruments Using Process Technology:* Data collection instruments are created by domain experts using a process-aware configurator component ⓐ. The latter provides an abstract and comprehensible modeling notation for domain experts to specify the flow logic of the mobile data collection instrument. Navigation operations as well as the data elements of instruments are modeled. Data elements, in turn, are connected to pages. Note that the latter are important for rendering instruments as they represent single screens on the smart mobile device and allow thematically structuring a questionnaire. In the context of questionnaire instruments, data elements represent questions, whereas navigation allows skipping questions (or even pages) depending on previously given answers. Finally, the configurator component allows defining rules for the automated evaluation of gathered data ⓑ.

2) *Generate Mobile Applications Based on Process Models:* The process model of a data collection instrument is used to drive its execution on the various mobile operating systems. In turn, this required the implementation of a generic mobile process engine. By interpreting process models directly on smart mobile devices, all changes of instruments can be realized in an easy and cost-efficient manner. Note that this allows for flexible mobile data collection applications. Furthermore, instruments are rendered locally on the smart mobile devices. The rendering mechanism, in turn, takes different mobile operating systems as well as screen sizes into account, again utilizing information from the process model.

3) *Relieve IT Experts through Automatic Process Management:* As depicted in Fig. 4, the process model ① as well as the analysis rules ② are mapped to XML documents. The latter are then automatically deployed to the respective smart mobile devices ③. Log files capturing execution information are stored using an XML structure to allow for their subsequent evaluation ④. In this context, security ⑤ is ensured based on state-of-the-art data encryption techniques. Note that the communication required for steps ① – ⑤ relies on Web Services [10]. Based on this automation, many challenging requirements of mobile data collection application projects are mitigated. For example, when releasing new versions of already existing instruments, IT experts are no longer required. Note that release management constitutes the main cost driver in the context of the discussed mobile data collection projects (cf. Section I). Finally, changes

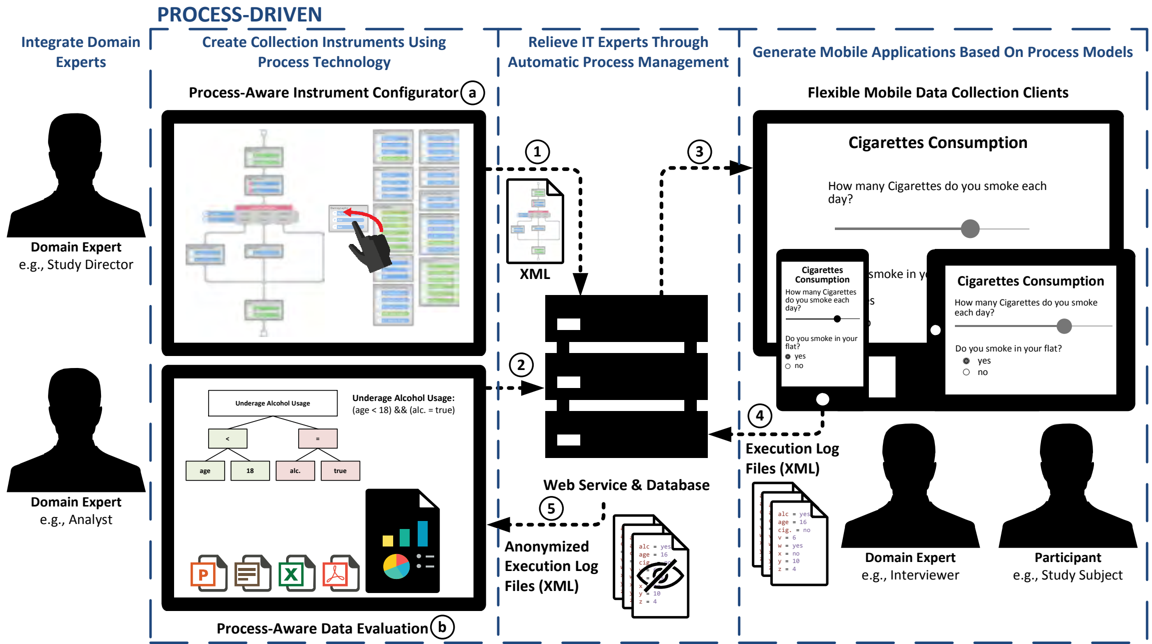

Figure 4. Architecture for Supporting Flexible Mobile Data Collection

solely affecting the XML documents require implementation adaptations to be performed by IT experts. For example, new legal regulations may cause changes of the used data encryption algorithm with respect to the XML document.

IV. Empowering Domain Experts

The fundamental goal of the configurator component is to empower domain experts to realize data collection instruments at a high level of abstraction and in a flexible way (cf. Fig. 5). The configurator is implemented as a Java Eclipse RCP application (cf. Fig. 5, a) and is connected to an intermediary service (cf. Fig. 4, 1 – 5). The latter communicates with mobile applications (cf. Fig. 5, c). The configurator component cannot be presented in detail. Its major functions are as follows:

1) **Modeling Area View:** The modeling area covers four aspects. First, the data collection instrument is captured and visualized as a process model (cf. Fig. 5, 1). Second, easy-to-use operations (i.e., drag & drop) for inserting and deleting pages are provided. Third, operations for adding and deleting gateways are provided. Gateways, in turn, allow for a sophisticated navigation within data collection instruments. Finally, it is required that all parameters of gateways are specified. Hence, advanced wizards are introduced in order to ensure that all required parameters are set.
2) **Page Repository View:** The repository covers three aspects (cf. Fig. 5, 2). First, it lists all available pages that may be used when composing the data collection instrument in the modeling area. Second, the version of a designed page may be selected. Recall that versioning constitutes a key requirement for domain experts. Third, it allows moving pages (i.e., respective version) to the modeling area via drag & drop.
3) **Element View:** The element view enables domain experts to create elements for pages (cf. Fig. 5, 3; only a small part of the entire view is shown). The configurator component provides five basic element types (cf. Table II) divided into two categories: Elements of the first category are solely used to visually structure a page, while the second category comprises elements dealing with data collection. In general, a domain expert needs to specify nine attributes when creating a new element: ID, question, language, question type (cf. Table II, 4), style information (e.g., alignment; optional), question mode (mandatory or optional), anonymization (optional), version information, and pre-defined answers (optional). Note that the page view, which is not shown in Fig. 5, is connected to the element view. Finally, all created elements may be used to model questionnaire pages via drag & drop operations across the two views.
4) **Preview Mode:** The preview mode can be easily accessed from all views and allows previewing elements or pages. Domain experts may configure three properties. First, a concrete smart mobile device (e.g., iPhone 6S) may be selected. Second, the screen orientation (i.e., landscape or portrait) may be specified. Third, the language to be displayed (e.g., French) may be selected. The specified information is then used to dynamically render a preview of the element as if it would be displayed on a real device. Note that such feature is highly welcome by domain experts.

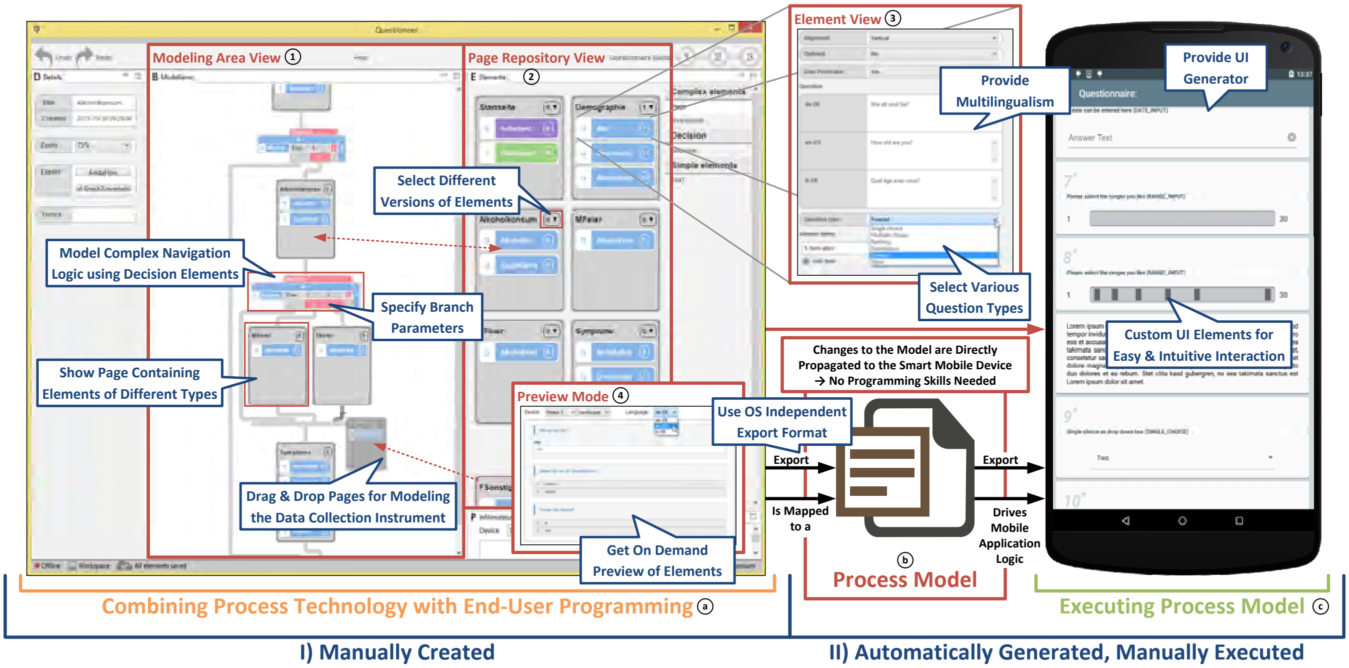

Figure 5. Empowering Domain Experts With Mobile Data Collection Through End-User Programming

Data Collection Explanation Elements	
1. Headline	Introduces the following elements.
2. Text	Provides additional information to assist users.
3. Media	Provides additional media information (e.g., images) to assist the user.
Data Collection Elements	
4. Question Types	
4.1. DropDown	Only one item may be selected.
4.2. Single Choice	Only one item may be selected.
4.3. Yes No Switch	Only one item may be selected.
4.4. Range	Multiple items may be selected.
4.5. Multiple Choice	Multiple items may be selected.
4.6. Ranking	Items may be ordered according own preferences.
4.7. Distribution	Points may be spent among available items.
4.8. Slider	One value from a predefined range may be selected.
4.9. Freetext	Answer using regular text input (text, number, date).
5. Sensor Types	
5.1. Camera	Take pictures during data collection.
5.2. Microphone	Record audio during data collection
5.3. Pulse Sensor	Measure the pulse rate during data collection.

Table II
ELEMENTS USED FOR FLEXIBLE MOBILE DATA COLLECTION

After modeling a data collection instrument, it is deployed to the intermediary service. The respective procedure, in turn, includes the automatic mapping of created data collection instruments to process models. The latter can then be deployed to the smart mobile application ⓒ, which consists of a lightweight process engine capable of dynamically interpreting process models; i.e., the engine controls and monitors the execution of questionnaire instances (e.g., automatically selecting branches at gateways depending on previous answers). Note that all changes applied to the data collection instrument can be made immediately available on the smart mobile devices when applying this model-driven approach and using the lightweight process engine. Based on this approach, changes to data collection instruments no

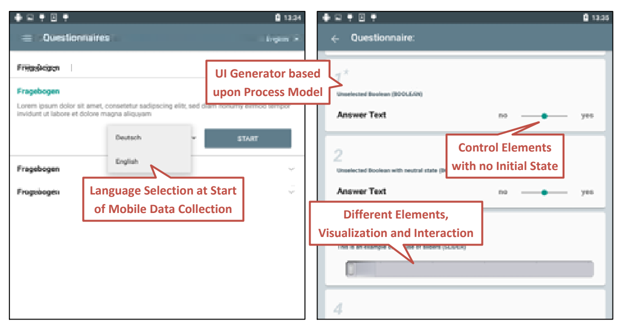

Figure 6. Smart Mobile Data Collection Application

longer require the involvement of IT experts. Furthermore, a sophisticated layout generator is provided to build and render the user interface automatically. In this context, platform-specific design guidelines were considered (cf. Fig. 6), which introduce novel control elements to meet domain-specific requirements (e.g., switch controls with no initial state).

V. PRELIMINARY RESULTS

Recall the fundamental facets from Fig. 1 and the technical glue for integrating them in a proper way (i.e., a process-driven approach) from Fig. 4. When realizing the technical solution, two aspects were of particular importance (cf. Fig. 5, I & II):

- A1 The paradigm for modeling data collection instruments needs to be accepted by domain experts (cf. Fig. 5, I).
- A2 IT experts are relieved from deploying data collection instruments to smart mobile devices (cf. Fig. 5, II). An appropriate approach enabling domain

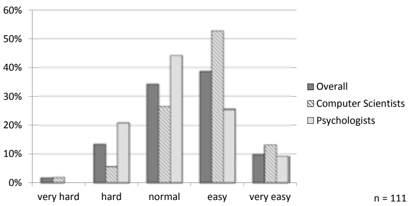

Figure 7. Perception of Mental Effort during Modeling

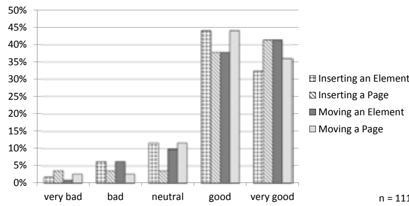

Figure 8. Perception of Mental Effort when using Basic Operations

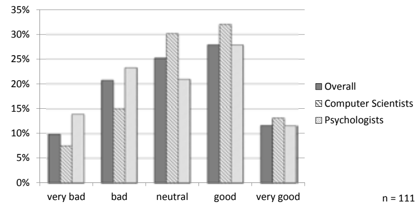

Figure 9. Perception of Mental Effort when using Complex Navigation Operations

experts to address advanced requirements (e.g., flexible changes) is provided.

Regarding A1, preliminary results of a study with 111 subjects are presented. The latter were either *Computer Scientists* (48%), *Psychologists* (39%), or *Others* (e.g., Business Scientists; 13%). 57 subjects were male (51%), whereas the other 54 subjects were female (49%). Subjects age ranged from 19 – 62 years, with an average age of 29 years. The study was conducted by four steps:

- S1 The configurator component was introduced. In this context, the main concept of modeling data collection instruments as well as basic operations (e.g., inserting pages) were briefly presented.
- S2 A paper-based instrument was provided to subjects, who then should realize it using the presented configurator component.
- S3 Subjects had to cope with *change requests* regarding an already existing data collection instrument (e.g., changing the order of elements or pages).
- S4 A survey comprising questions related to the user interface, the usability of the configurator, and the change operations available was presented.

In the following, selected study results are presented. *First*, Fig. 7 summarizes how subjects perceived the complexity of steps S1 – S3. On the one hand, the discrepancy between Computer Scientists and Psychologists was rather low. On the other, the main part of the subjects considered the overall complexity of modeling a data collection instrument as *normal* or better. *Second*, Fig. 8 illustrates how subjects perceived the complexity of the basic operations applied in the context of steps S1 – S3. Overall, the feedback was positive. Most of the subjects were able to correctly and properly apply the operations. Furthermore, they considered their usage as intuitive. *Third*, Fig. 9 indicates the mental effort perceived by the subjects when applying complex navigation operations during steps S1 – S3. As illustrated, the perceived mental effort was high for the majority of the subjects. Consequently, further research on usability issues is needed, e.g., to improve user explanations. Finally, the study compared modeled instruments with a reference model designed by experts. Note that 81% of the models provided by the subjects were sound; 64% of the psychologists submitted a correct model.

Regarding A2, algorithms for transforming a given data collection instrument into a process model, which may then be executed on smart mobile devices, are required. In this context, process models are represented in terms of an XML structure. Fig. 10 presents algorithms along the entire transformation procedure. Due to lack of space, only the main algorithm mapping the data collection instrument to the process model (cf. Alg. 1) is presented. Note that the algorithm ensures that the processed XML structure can be correctly executed on the smart mobile devices. For example, it automatically detects data dependencies violated by a modeler. Since domain experts may create complex mobile data collection instruments, the algorithm is structured into sub-algorithms. For example, Alg. 2 is invoked to evaluate whether a modeled page can be correctly inserted into the process model. Furthermore, data elements that are needed to store the respective answers will be automatically assigned to this model as well.

Altogether, the realized algorithms worked properly. In future work, additional issues will be considered. For example, further research is needed to provide enhanced features enabling complex navigation. Again, correctness issues have to be carefully considered when applying changes to the already introduced algorithms in order to properly support domain experts in this regard.

Algorithm 1: Traversing the Graph of the Data Collection Instrument and Calling Mapping Methods

```
Data:
handler: Implementation for the export interface
cElement: Current element to be processed
begin
    if (cElement instanceof QPage) then
        handler.handlePage(cElement) ;  // map element (page) to node (cf. Algo 2)
        traverseGraph(handler, cElement.getNext()) ;  // and continue with next element
    else if (cElement instanceof QConnector) then
        handler.handleConnector(cElement) ;   // map element (connector) to an edge
        /* do not continue with the next element if it is a join node.
           Join gateways are handled later                                  */
        if !(cElement.getNext() instanceof QJoin) then
            traverseGraph(handler, cElement.getNext());
        end
    else if (cElement instanceof QSplit) then
        handler.handleSplit(cElement) ;          // map element (split) to a gateway
        traverseGraph(handler, cElement.getNext());
        // now handle all paths of the gateway (→ connector)
        foreach path in cElement.getNext() do
            traverseGraph(handler, path);
        end
        // and handle the join node immediately after
        traverseGraph(handler, cElement.getCorrespondingJoin());
        /* continue with the element (connector) right after the join
           element, as it would get called several times (for each path)
           when handling it later                                           */
        traverseGraph(handler, cElement.getCorrespondingJoin.getNext())
    else if (cElement instanceof QJoin) then
        handler.handleJoin(cElement) ;           // map element (join) to a gateway
    else if (cElement instanceof QStart) then
        handler.handleStart(cElement);
        traverseGraph(handler, cElement.getNext());
    else if (cElement instanceof QEnd) then
        handler.handleEnd(cElement);
        traverseGraph(handler, cElement.getNext());
    else
        // An unsupported type of element found → Exception-Handling
    end
end
```

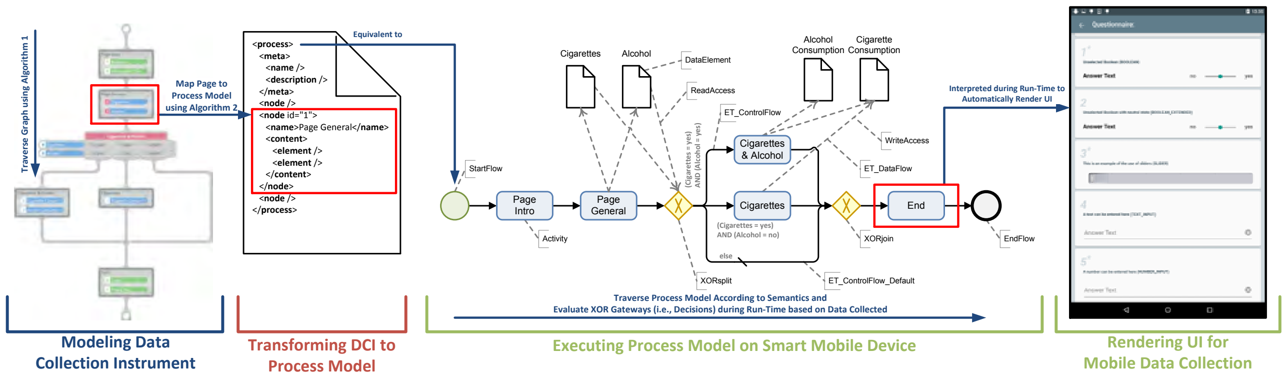

Figure 10. Mapping a Data Collection Instrument to a Process Model for Mobile Data Collection

Algorithm 2: Mapping a Page (Data Collection Instrument) to a Node (Process Model)

```
Data:
cElement: Current element to be mapped
model: The process model where elements should be inserted
pred: The predecessor of the new node (e.g., pred = startnode in the first run)
succ: The successor of the new node (e.g., succ = endnode in the first run)
begin
    // Map the current element to a node
    // (e.g., copy its name and content to a node object)
    ProcessNode node = new ProcessNode(cElement);
    // check if the new node can be inserted between the given nodes
    if (Operator.checkNode(model, node, pred, succ)) then
        Operator.insertNode(model, node, pred, succ); // Insert the node between them
        foreach (CElement element in cElement.getElements()) do
            // Iterate over all Elements inside the Page
            if (element instanceof ElementQuestion) then
                // Add a DataElement to save the Data collected
                ProcessDataElement pde = new ProcessDataElement(element);
                // Assign the DataElement to the node with connector
                node.assignDataElement(pde, ProcessDataElement.ACCESS_WRITE);
            end
        end
        pred = node; // set pred to the node just inserted before
    else
        // It is not possible to insert the node → Exception-Handling
    end
end
```

VI. Related Work

Two categories of related work are relevant in the context of this paper.

A. Approaches Dealing with End-User Programming

In the past, various approaches supporting non-programmers with creating software were suggested [11], [12]. Both their feasibility and applicability were proven in a multitude of studies. For example, [11] provides an environment assisting system administrators in their daily routines and allowing for the visual modeling of script applications. A related case study revealed that the administrators were able to easily create the scripts needed.

[13] introduces a notation that may be used to visually implement simple programs based on blocks. Thereby, each block represents a function of the final computer program. Similar to the presented approach, provided blocks may be moved using drag & drop operations. Evaluations with pupils showed that they strongly prefer this approach compared to traditional text-based programming. Furthermore, teachers reported that the simplified representation of programs significantly improved the basic understanding. In turn, [14] provides a sophisticated plug-in architecture on top of the first discussed approach that enables non-programmers to manipulate and extend the modeling notation. Again, the feasibility could be proven by the authors. [15] presents end-user programming approaches for creating Web Mashups. Users may add data sources and apply processors (e.g. operators and functions) to modify respective data in a graphical editor.

B. Approaches Dealing with Mobile Data Collection

A platform supporting researchers with collecting data using smart mobile devices is presented in [16]. However, this platform is specifically tailored to mental health research and cannot be easily adapted to other domains. Furthermore, [16] deals with questionnaire customization. In particular, interval-based interviews are provided and data from external sensors may be integrated. However, the approach does not provide features like automatic UI generation. [17] presents a framework for mobile data collection as well. Similar to the presented approach, a configurator component is provided with limited features compared to the ones presented in this paper. For example, only few elements for structuring mobile data collection instruments are available. [18] provide benefits of using smart mobile devices in healthcare and introduces various *mHealth* applications.

VII. Conclusion and Future Work

Based on the experiences with implementing real-world mobile data collection applications, major challenges associated with the described facets were elaborated. Along these facets, the idea of a generic mobile data collection framework was introduced. In particular, the framework enables domain experts to create sophisticated mobile data collection applications on their own. Using techniques known from end-user programming provides promising perspectives regarding this challenging endeavor. Furthermore, a process-driven approach has proven its appropriateness for automatically transferring mobile data collection applications to smart mobile devices in a flexible and robust way. In particular, as shown with the configurator component, modeling processes is a feasible method for creating mobile data collection applications by domain experts. Process technology delivers features to meet the requirements of the domain experts. In turn, technical challenges could be tackled by using this technology directly on the smart mobile device. Altogether, combining process management technology with end-user programming indicates first promising results for a generic mobile data collection framework.

Although the configurator component was welcome by the domain experts, further research is needed with respect to usability issues. For this purpose, another study will be conducted applying methods from usability engineering [19] to the presented configurator components. Furthermore, features enabling domain experts to analyze collected data is another promising research direction. In this context, process mining algorithms may be applied to collected data in order to obtain more valuable insights [20]. In addition to the already introduced results with respect to mental effort, more performance indicators of the framework need to be evaluated. The latter include modeling time for a questionnaire as well as performance metrics compared to paper-based instruments.

Moreover, the proposed approach may significantly change the way of creating mobile data collection applications in other application domains as well. In particular, life sciences may benefit as they mainly focus on everyday life context situations that can only be properly covered when using complex data collection applications.

REFERENCES

[1] R. Pryss, M. Reichert, B. Langguth, and W. Schlee, "Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research," in *IEEE 4th Int'l Conf on Mobile Services*. IEEE Computer Society Press, June 2015.

[2] M. Ruf-Leuschner, R. Pryss, M. Liebrecht, J. Schobel, A. Spyridou, M. Reichert, and M. Schauer, "Preventing further trauma: KINDEX mum screen - assessing and reacting towards psychosocial risk factors in pregnant women with the help of smartphone technologies," in *XIII Congress of Europ Soc of Traumatic Stress Studies Conf*, June 2013.

[3] J. Schobel, R. Pryss, and M. Reichert, "Using Smart Mobile Devices for Collecting Structured Data in Clinical Trials: Results From a Large-Scale Case Study," in *28th IEEE Int'l Symp on Computer-Based Medical Systems*. IEEE Computer Society Press, June 2015.

[4] S. Wilker, A. Pfeiffer, S. Kolassa, T. Elbert, B. Lingenfelder *et al.*, "The role of fkbp5 genotype in moderating long-term effectiveness of exposure-based psychotherapy for posttraumatic stress disorder," *Translational Psychiatry*, vol. 4, no. 6, 2014.

[5] D. Isele, M. Ruf-Leuschner, R. Pryss, M. Schauer, M. Reichert, J. Schobel, A. Schindler, and T. Elbert, "Detecting adverse childhood experiences with a little help from tablet computers," in *XIII Congress of Europ Soc of Traumatic Stress Studies Conf*, June 2013.

[6] R. Pryss, N. Mundbrod, D. Langer, and M. Reichert, "Supporting Medical Ward Rounds Through Mobile Task and Process Management," *Information Systems and e-Business Management*, vol. 13, no. 1, February 2015.

[7] M. Reichert and B. Weber, *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Berlin-Heidelberg: Springer, 2012.

[8] J. Schobel, M. Schickler, R. Pryss, and M. Reichert, "Process-Driven Data Collection with Smart Mobile Devices," in *10th Int'l Conf on Web Information Systems and Technologies*, ser. LNBIP. Springer, 2015, no. 226, pp. 347–362.

[9] M. Reichert and P. Dadam, "Enabling Adaptive Process-aware Information Systems with ADEPT2," in *Handbook of Research on Business Process Modeling*, J. Cardoso and W. M. P. van der Aalst, Eds. Hershey, New York: Information Science Reference, March 2009, pp. 173–203.

[10] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson, *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. Prentice Hall PTR, 2005.

[11] E. Kandogan, E. Haber, R. Barrett, A. Cypher, P. Maglio, and H. Zhao, "A1: End-User Programming for Web-based System Administration," in *Proc 18th ACM Symp on User Interface Software and Technology*. ACM, 2005.

[12] E. Klopfer, S. Yoon, and T. Um, "Teaching complex dynamic systems to young students with starlogo," *Jour of Computers in Mathematics and Science Teaching*, vol. 24, no. 2, pp. 157–178, 2005.

[13] A. Begel and E. Klopfer, "Starlogo TNG: An Introduction to Game Development," *Jour of E-Learning*, 2007.

[14] R. V. Roque, "OpenBlocks: An Extendable Framework for Graphical Block Programming Systems," Master's thesis, Massachusetts Institute of Technology, 2007.

[15] J. Wong and J. I. Hong, "Making Mashups with Marmite: Towards End-user Programming for the Web," in *Proc of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2007, pp. 1435–1444.

[16] A. Gaggioli, G. Pioggia, G. Tartarisco, G. Baldus, D. Corda, P. Cipresso, and G. Riva, "A mobile data collection platform for mental health research," *Personal and Ubiquitous Computing*, vol. 17, no. 2, 2013.

[17] S. Kim, J. Mankoff, and E. Paulos, "Sensr: Evaluating a Flexible Framework for Authoring Mobile Data-Collection Tools for Citizen Science," in *Proc of the 2013 Conf on Computer Supported Cooperative Work*. ACM, 2013.

[18] D. D. Luxton, R. A. McCann, N. E. Bush, M. C. Mishkind, and G. M. Reger, "mHealth for Mental Health: Integrating Smartphone Technology in Behavioral Healthcare," *Professional Psychology: Research and Practice*, vol. 42, no. 6, pp. 505–512, 2011.

[19] J. Nielsen, *Usability engineering*. Morgan Kaufmann, 1994.

[20] W. M. P. van der Aalst and A. Weijters, "Process mining: a research agenda," *Computers in industry*, vol. 53, no. 3, 2004.