From Paper-Based to Mobile Checklists - A Reference Model

Thomas Boillat[1], Christine Legner[1]

[1]University of Lausanne, Faculty of Business and Economics (HEC), Lausanne, Switzerland
{thomas.boillat, christine.legner}@unil.ch

Abstract. Checklists are cognitive tools that ensure quality, safety and reduce human errors when executing working routines. Besides their popularity in practice, checklists increasingly receive attention from academics, who have even called for a "science of checklists". According to prior studies, mobile checklists are more effective than their paper-based alternatives, but research in this domain is still limited. It focuses on mobile checklists' use and benefits, rather than on their characteristics and design. Our study aims at developing a reference model for conceptualizing mobile checklists. The suggested reference model has been constructed by following design science principles, based on an extensive analysis of paper-based and mobile checklists from the literature and from the practical world. Its main contribution is a shared understanding of the domain knowledge between users and developers, which helps to communicate innovative ideas about mobile checklist applications.

Keywords: checklists, mobile applications, reference model, organizational routines, cognitive tools

1 Introduction

Checklists are cognitive tools that make people work better. Companies have embraced the use of checklists in many fields because of their effectiveness and ability to reduce human errors [1]. Checklists are of particular interest whenever quality and time must be respected, which explains their widespread use in activities such as inspection, maintenance and installation. Besides their popularity in practice, researchers in medical, engineering, and other domains have studied (paper-based) checklists as cognitive tools and elaborated on their use since the 1980s. More recently, there have been calls for a "science of checklists" to advance the development, implementation and evaluation of checklists [2]. With the increasing proliferation of mobile devices in business environments [3], many activities traditionally performed with paper copies are now realized by means of smartphones and tablets. Mobile checklists, which are accessible on smartphones and tablets, provide new ways of visualizing checklists on colorful and tactile screens, and of filling them with virtual keyboard or hand gestures. They are likely to overcome some of the known limitations of paper-based checklists that are distributed as paper copies and filled in using a pen during execution. According to first empirical studies, the use of mobile devices as a support

12th International Conference on Wirtschaftsinformatik,
March 4-6 2015, Osnabrück, Germany

to checklists brings more satisfaction and reduces the number of errors [4]. Nevertheless, existing research on mobile checklists is still limited and focuses on their use and benefits, rather than on their characteristics and design. While we observe an increasing number of mobile applications that implement checklist functionality, we still lack a broader understanding of how to conceptualize and design effective checklists in general [2] and specifically for mobile devices. Against this gap in research, this paper aims at developing a reference model for mobile checklists. We followed design science principles to build our reference model and relied on literature and an extensive analysis of 22 paper-based and mobile checklists. The reference model has been subsequently evaluated by experts and refined based on their input. As a prescriptive artifact, the suggested reference model provides the basic vocabulary and conceptualization to support IS design and help to communicate innovative ideas on mobile checklist applications.

The remainder of this paper is organized as follows: We start by reviewing prior research related to the role of checklists as cognitive tools, their structure and electronic representation. Section 3 is dedicated to our research approach and the reference model's construction and evaluation cycles. In the section 4, we analyze existing paper-based and mobile checklists, before we elaborate on the suggested reference model in section 5. Section 6 presents its evaluation and section 7 discusses our findings.

2 Prior Work

2.1 Checklists as Cognitive Tools

Checklists are common cognitive tools that can help complete a task as simple as shopping or as complex as flying a Boeing 747 [2]. Checklists are considered as the instantiation of a procedure in a physical form [2] and standardize knowledge by means of a sequential set of steps [5]. Checklists are recognized tools to democratize knowledge and reduce human errors, while they reduce workload, improve quality, communication, and collaboration [2, 6, 7]. They can protect against memory laps, promote attention to thoroughness, serve to inform about changes in standards of care, and build a cohesive team that together can outperform a single individual [8]. The popularity of checklists is also due to their ease of use [5], which allows less skilled or less trained people to execute complex tasks. Checklists have the ability to effectively translate abstract process descriptions into actionable procedures while providing task documentation to support their execution [2]. Checklists are particularly important in emergency situations, when quality and time must be respected, and pressure is extremely high. But also in routine situations when the sequential execution is critical, when competing priorities distract attention and when the knowledge between executants is variable [2]. Aviation and nuclear plants are considered pioneers in using paper-based checklists [1], while health care increasingly relies on checklists to reduce human errors [2].

2.2 Structure and Types of Checklists

From a conceptual point of view, a checklist consists of a series of linked tasks that need to be completed within a certain amount of time and in a specific sequence [5]. Checklists need to be distinguished from simple To Do lists, which comprise lists of tasks that are neither linked nor of the same nature. In the case of checklists, the sequential execution of tasks is often considered as critical. Additional checks are therefore introduced to ensure that each task is completed prior to executing the next one [2, 5]. The sequences can be structural or functional [5]. In the first case, the sequence follows the physical structure of the object treated by the checklist. For instance, the executant would check all the pieces of a wing during an aircraft inspection. In the second case, the sequence follows a specific function: the executant would inspect the entire break system, and then the engine. Alternatively or additionally, a sequence can follow a specific timeline (e.g., 30 minutes before, 15 minutes before a certain event).

Besides the characteristics mentioned above, existing studies have not paid much attention to describing the structure of checklists, but focus more on processes and guidelines for creating effective checklists in specific domains. For instance, a study in medical practices argues that an effective checklist should provide 'unambiguous guidance on what, when, how, and who should do the interventions and should be logistically efficient and easily performed' [2]. Certain domains – e.g., aviation – distinguish two different types of checklists [7]: one for normal operations and another for emergency situations. The latter contains a larger number of instructions with more detailed descriptions, but also branches used as scenarios to better fit with the situation. Checklists can also be defined according to their execution types. Based on the number of people involved in performing and verifying the action and the configurability of tasks, [2] define four principal types of checklists: static parallel (when a checklist is completed and executed by one operator as a series of read-and-do tasks); static sequential with verification (when one operator reads a series of tasks and a second person verifies that each task has been correctly completed); static sequential with verification and confirmation (used within teams, when one person reads the tasks and each responsible person verifies the completion of his or her specific tasks); and dynamic (used to guide complex decision-making, when the checklist has different options that are defined by an algorithm).

2.3 State of the Art on Electronic and Mobile Checklists

Electronic and mobile checklists address certain shortcomings that come along with paper-based checklists, such as the accuracy of data [5] or the difficulties associated with carrying and maintaining paper copies [2]. At the same time, electronic checklists may provide more comfort and increase effectiveness [1]. However, the first attempts were not always promising. In 1985, a study conducted in the nuclear power domain revealed that the use of electronic support got the workers lost in their procedures due to the difficulty to navigate between and within the procedures [9]. In 1998, a study in aviation, a pioneer domain that has been extensively using checklists,

showed that electronic procedures were easier to learn and to use and were faster and more accurate than traditional paper-based procedures [10]. For the same domain, electronic checklists were found to reduce the number of tasks skipped, to reduce time required to keep track when coming back on incomplete checklists and to increase readability [7]. In 2000, some researchers saw an opportunity to use PDAs (Personal Device Assistants) as an answer to existing problems with paper-based checklists – e.g., workers loosing track in endless checklists, skipping tasks and using old versions of checklists [5]. More recently, mobile checklists for OSCE (Objective Structured Clinical Examinations), were found to provide a better comfort of use and also reduce the number of unchecked (skipped) elements [4]. Interestingly, examiners tend to change their answers more often than with paper copies. The study argues that electronic support facilitates the modification of answers without altering the clarity and visibility of the answers (no need to erase or use correction fluid).
From the first electronic checklist applications in 1985 to now, technology has made tremendous progress. Though existing research provides empirical evidences that electronic and mobile checklists perform better than paper-based checklists, there has been little research on the characteristics and structure of such checklists. We consequently lack a more thorough understanding about how to design effective checklists [2]. Since there are no general conceptualizations of mobile checklists, new checklists are created from scratch, without reusing existing knowledge. This does not only slow down the implementation process, but also increases the upfront effort to develop mobile checklists and has negative impact on efficiency and effectiveness.

3 Research Approach

Against the aforementioned research gap, our goal is to design a reference model of mobile checklists that may guide researchers and practitioners in designing effective checklists. As prescriptive knowledge, reference models describe and explain the standard decomposition of a known problem domain into a collection of interrelated parts, or components, that cooperatively solve the problem [11, 12]. They are documented by means of semi-structural languages [13] and allow one to accelerate the development of information systems, reduce the corresponding costs, help to communicate innovative ideas and best practices, and reduce the risk of failure [14].

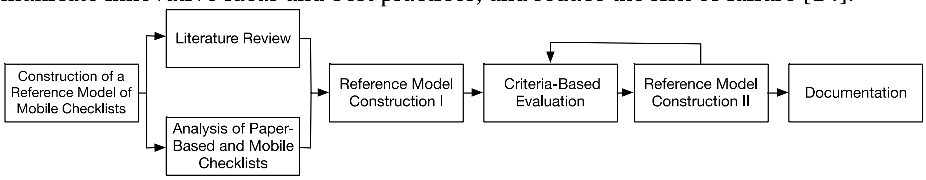


Fig. 1. Construction and Evaluation Process

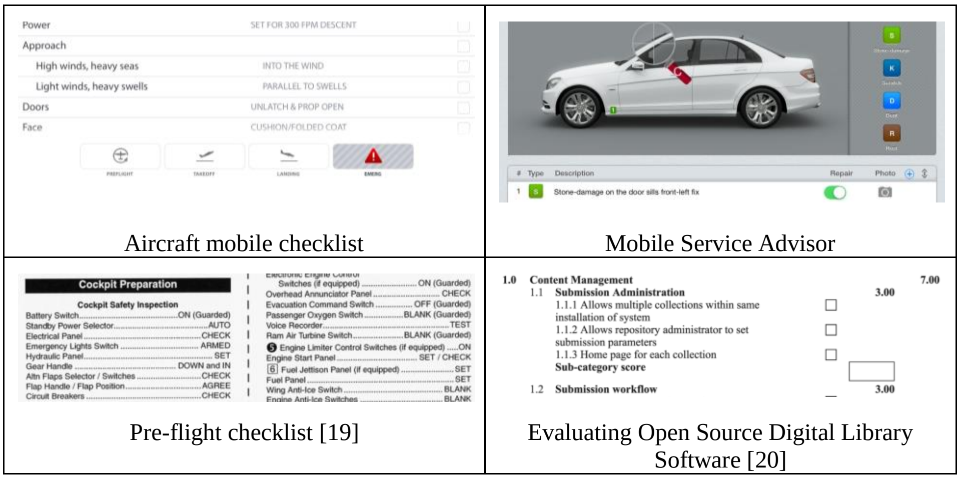
The design of reference models as artifacts follows the principles of design science, with iterative construction and evaluation cycles [11]. In our case (see Figure 1), we relied on an inductive approach to construct the reference model, making use of a literature analysis and a detailed investigation of artifacts from the real world. We first identified scientific literature based on the key words structure, characteristics and design of checklists, resulting in more than 220’000 references on Google Scholar. We realized that these keywords were used in many kinds of research, but that only a very small number of papers investigate checklists from a conceptual perspective. Then, from five well-targeted papers, we performed reverse citation research and found another three papers. To complement the literature, we investigated checklist artifacts from the practical world. Our sources for identifying checklists were scientific literature as well as practitioner journals and app stores. In order to ensure that our reference model is robust and covers typical checklists, but also communicates innovative ideas, we aimed for a broad representation of different application areas as well as coverage of both, traditional paper-based and innovative mobile checklists. In total, we analyzed 22 paper-based and mobile checklists in various fields (i.e., inspection, maintenance and installation) and industry domains (e.g., health care, aviation, field service). Out of the 22 checklists, 12 were implemented as paper-based checklists, and 10 were in the form of mobile checklists, except one electronic checklist in the case of an embedded system in an aircraft. Various fields were represented; health care (6 checklists), aviation (3), field service (2), software engineering (2) as well as generic mobile checklists for inspection (5) and event organization and safety. Table 3 in the Appendix lists the 22 checklists with their characteristics. For each checklist, we systematically analyzed and described its structure and characteristics to represent the artifact as a reference model. In the case of paper-based checklists, we used the paper documentation as main source of information, whereas mobile checklists were downloaded from the Apple Store and then tested on an Apple iPad. We analyzed the resulting 22 models in order to identify commonalities and generalize them into a reference model.

The evaluation of a reference model plays an important role to ensure and improve the model’s quality [12]. The quality of a model directly impacts the implementation time and costs, if some modifications on the model are required to provide the expected functionalities [12, 15]. There is no standard when it comes to evaluating the quality of models, due to the level of interpretation that comes along with the activity of modeling [16]. In order to evaluate our reference model, we selected the framework of Lindland et al. [15], which is widespread and well accepted in the field of conceptual and reference modeling [12] and was empirically validated [17]. We assume that its notoriety will facilitate the evaluation, at least with scholars. Based on this framework, we evaluated the suggested reference model in two rounds of guided expert interviews with regard to its syntactic, semantic and pragmatic quality (based on [15], detailed criteria see Table 2 in Section 6). Based on these iterative construction and evaluation cycles, we derived the final version of the reference model.

4 Analysis of Mobile and Paper-Based Checklists

Our analysis covers 22 paper-based and mobile checklists of different complexity and size. The average number of tasks included in the checklists analyzed was 50, while the shortest checklist included 11 tasks and the most comprehensive one 122.

By *paper-based checklists* we understand checklists that are distributed as paper copies and filled in using a pen during execution. In our sample, we included paper-based checklists that are widely recognized for their effectiveness such as the *Surgical safety checklist*, which was empirically demonstrated to reduce mortality and complication rates. Other checklists, such as *Evaluating digital library software*, which relies on weighted tasks to evaluate libraries, were selected for their interesting structure. Compared to paper-based checklists, *mobile checklists* run on mobile devices, such as smartphones or tablets, and can be filled using a virtual keyboard or hand gestures. Among the investigated mobile checklists were innovative checklists such as the *Mobile Service Advisor*, which received a prestigious European app challenge award [18]. We also selected checklists (i.e., pre-flight checklists) available in multiple versions, paper-based and mobile, in order to analyze commonalities and differences. Among them were mobile checklists that have been subject of scientific studies, such as the *electronic Objective Structured Clinical Examination (eOSCE)*, which demonstrated better ease of use than its paper-based alternative.



Aircraft mobile checklist | Mobile Service Advisor

Pre-flight checklist [19] | Evaluating Open Source Digital Library Software [20]

Fig. 2. Sample checklists - Mobile checklists on top, paper-based on bottom

As analysis framework, we relied on four key characteristics of checklists mentioned in prior literature:

1. Prior literature suggests that checklists provide unambiguous guidance on what, when, how, and who should execute a task. We therefore analyzed the *documentation of tasks* in more detail: How are tasks described and explained?
2. Since the links between tasks and their sequence is considered critical, we focused on analyzing how checklists organize these *links between tasks*: How are tasks logically linked and dependent on each other (hierarchy and sequence)?
3. Checklists should standardize task execution with the goal of reducing errors. We consequently investigated how checklists *monitor and control execution*: how do checklists support monitoring and controlling task execution (verification and validation)?

4. Finally, checklists may be used by individuals and in teams. We therefore analyze the *role and type of executants and interveners*: Which roles are using the checklists?

When it comes to documenting tasks (1), paper-based checklists merely describe the tasks as short, plain text due to the limitations of paper documentation. In some cases, pictures, references or contacts are added to access additional information. We also observed that paper-based checklists often cover a large variety of procedures. For instance, the *how to organize an IEEE event checklist* was designed for any IEEE events and thereby informs the executants in which cases they should perform one task or another. Compared to their paper-based counterparts, mobile checklists leverage the technological capabilities of mobile devices to provide a much richer description of tasks, through the use of pictures and 3D-models, videos and links to external documents and web sites. The embedded capabilities can also serve to document executed tasks by means of picture through the mobile device's camera. In addition, task lists can be pre-configured based on the location or the object on which a checklist is carried out. An example is the *Mobile Service Advisor* checklist, where task lists for car inspections are created depending on the car model and history.

With regard to the logical links between tasks (2), we observed that tasks are typically divided in multiple groups to bring more structure to the checklists. We found both sequence structures, physical and functional. Examples of physical structure are the inspection of airplanes (*Pre-flight inspection checklist [19]*) during which the exterior is checked, and then the cockpit. The *Mobile Service Advisor* and the *Car inspection* rely also on a physical structure in which the inside, chassis and the bottom are checked. On the contrary, the *Surgical safety checklist* considers three functional groups that are *before the induction of anesthesia*, *before the skin incision* and *before the patient leaves.* We observed a functional structure also in the *Crane checklist, construction site* and *how to build a custom home* checklist. We also noticed that time–lines were used to indicate when to execute tasks, e.g. in the *How to organize an IEEE conference* checklist, where the first tasks start two years before the conference takes place and the last ones six months after it. While groups on paper-based checklists are static due to restriction of paper copies, mobile checklists sometimes offer the possibility to create or modify groups. This is the case of the *Mobile filed service* or the *Audit compliance*, which allow one to add, modify or remove tasks from groups.

Both paper-based and mobile checklists consist of elements that allow one to monitor and control the execution (3). In most paper-based checklists, checkboxes must be ticked, once the task is done. In few cases, such as [21], the executant can select a state from a list (i.e., not at all, a little, somewhat, very much). The investigated mobile checklists provide significantly more possibilities of status reporting, from scales (e.g., 1 to 5) for task completion, to lists (e.g., yes, partially, no) and free text (see Figure 3 #1). Often, when a task is marked as performed, the color changes to green or to red, when it is missed. We also noticed that mobile checklists inform more often about the expected task result than paper-based checklists. The pre-flight checklist [19], for instance, specifies task results, e.g. that the task *Electronic engine control* must display *ON* after being performed. In the case of the eOSCE mobile checklist

(Figure 3), which aims at evaluating the clinical performance and competence of medical students, the examiner reads questions from the mobile device and directly sees the expected answers (Figure 3 #2). Additionally, mobile checklists can provide error messages when a task is missed or inform about the remaining time or tasks before finishing a checklist (Figure 3 #3, 10:28 minutes and 11 tasks remain). Finally, signature by the executant or the supervisor may be used to validate the execution of a checklist. In the case of paper-based checklists, very few had this feature, while it was much more common on mobile checklists, relying on digital signature.

Fig. 3. eOSCE checklist [4] - Abstract

With regard to executants (4), our analysis revealed that most paper-based checklists are handled exclusively by one person. Only in very critical procedures, another person, seen as supervisor, is involved to verify and confirm the tasks performed, such as in the *pre-flight checklist* [7, 19] and in the *Patient safety in operating rooms*. However, involvement and roles of other people were rarely written on the checklists, but only in their documentation. From a design point of view, we observed no difference between a checklist that is designed for one or for two people. Mobile checklists are even more individual-centric. In the cases we analyzed, we did not find the involvement of other people, not even for the mobile alternative of the pre-flight checklist.

5 Reference Model

Table 1 summarizes the characteristics of paper-based and mobile checklists and sheds the lights on the commonalities and differences. From this comparison, we identified the main elements of the reference model (see last column of the table).

Table 1. Characteristics of Paper-based and Mobile Checklists

Characteristics from literature	**Paper-Based checklists**	**Mobile checklists**	**Reference model**
Task documentation	Mainly description of tasks as plain text (few lines). Sometimes few pictures or references to additional info and contacts.	Exhaustive explanations comprising text, pictures, and videos. Links to external (web) documents. Use of camera as evidence.	Task documentation
	Rather generic, little information is provided or little room for entering data.	Context-specific, adapt the number and content of tasks.	Context
Links between tasks	Many groups and sub-groups.	Few groups displayed as tabs. Possibility to create or modify groups of tasks.	Hierarchy
	Physical, functional, timeline.	Physical and functional.	Sequence

Monitor and control execution	Checklists' statuses mainly use checkboxes, in few cases a short list of options.	Checklists' statuses often rely on list of options, use of colors, information about the expected result, verification of the consistency.	Verification
	In few cases, validated by manual signatures.	In many cases, validated by electronic signatures.	Instance
Users of checklists	Executant, supervisor.	Executant.	Roles
Checklists description and goals	Little information (title, version, creation date).	Much information describing the goal, automatic reports, completion of checklists.	Checklists' description

The suggested reference model (Figure 4) comprises 14 entities that are presented below. The notation is inspired from the Unified Modeling Language (UML) class diagram, which provides an adequate level of detail to express the entities and their attributes as well as their relations [22]. In order to reduce the complexity, the conceptual model does not contain associative tables. To facilitate the description of the model, entities are written in *italic.*

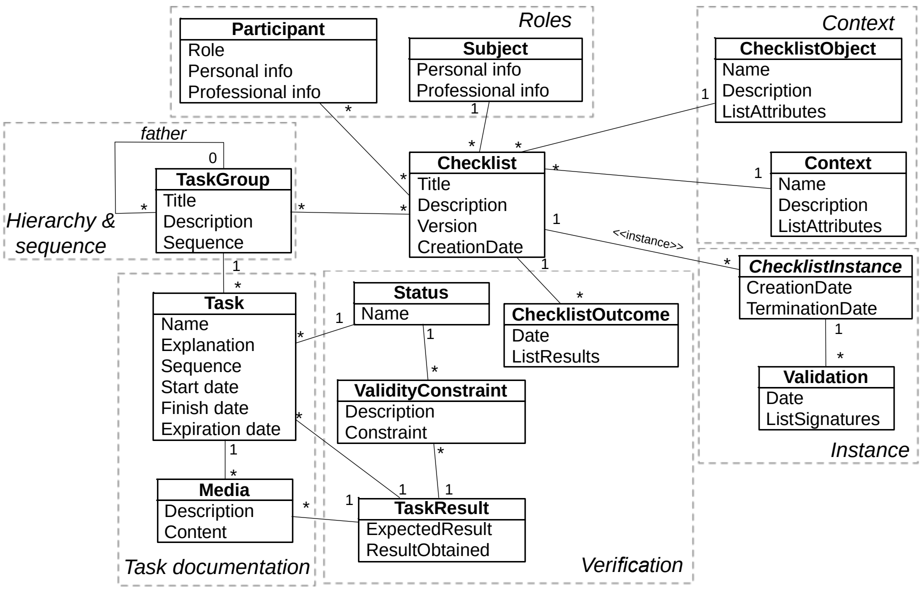


Fig. 4. Reference Model of Mobile Checklist

At the center of the reference model is the *checklist*, which connects all other entities. As described in the literature and section 4, a checklist consists of linked tasks that document and guide the execution. We consequently designed two groups of entities, hierarchy and sequence, as well as task documentation. In the first group, *taskGroup* aims to separate and sequence tasks in groups. It contains a name, an explanation and a sequence to order the tasks within a group. The *task* entity is linked to the *media* in which different pictures, videos or other files that document the tasks are stored, such as the 3D-models of cars used in the *Mobile Service Advisor* or the pictures taken by the camera in *Health inspection* checklists. As for managing sequences, both groups and tasks reference their predecessor thereby creating node structures as in graph theory [23]. The group or task that has no predecessor is the first one, called the root node in graphs. Groups or tasks pointing to the same predecessor are performed in parallel.

In order to verify the execution of tasks, the table *status* is connected to *validityConstraint*, which allows for verifying if a status meets some requirements (e.g., task 2 is mandatory but has been missed). Similarly, a task result (*taskResult*) can receive some constraints to verify if the result obtained is similar to the expected result. In the group verification belongs the checklist outcome (*checklistOutcome*) that stores the results from executing the checklist. From this entity, reports can be generated. A checklist is associated with an object (*ChecklistObject*) that is treated or transformed by the checklist. Some checklists comprise additional information related to the *context*, the *subject* and the *participant*. The context provides information with regard to the location (e.g., building, room) where the checklist is executed. The subject is the worker who executes the checklist, while the participant(s) are people involved at some points in the execution of the checklist (e.g., inspector). We observed that the object as well as the context might impact the tasks in a checklist. The reference model consequently relates the checklist to the role (*subject*) performing the tasks, the object (*checklistObject*) treated by the checklist – such as the plane model and characteristics in the case of the *Aircraft checklist* – and the context (*context*) in which a checklist is executed – such as the physical location. Each time a checklist is used, a new instance (*checklistInstance*) is created. The latter refers to the version of the *checklist* to ensure the latest release is picked. The table *validation* allows the executant (s) of the checklist or the supervisor to sign and thereby confirm the work done.

6 Evaluation

The evaluation consisted in two rounds of guided interviews with experts. As outlined in section 3, we relied on a widely used evaluation framework for reference models [15] which suggests evaluation criteria inspired by quality categories from the semiotic theory: 1) *Syntactic quality* describes the quality of the model in terms of language constructs, and the formal syntax. It aims to ensure that the constructs are expressed correctly. 2) *Semantic quality* focuses on the capability of the model to describe the domain being modeled, in our case mobile checklists. Thus, it considers relations among statements and their meaning. 3) *Pragmatic quality* evaluates how well the model correspond its audience interpretation. From the model properties, which are provided for each of the three quality categories [15], we derived a set of questions that were given to the experts during the evaluation process (see Table 2).We performed two rounds of analytic evaluation based on experiments. For the first round, we selected four experts in modeling and software development in our university department. From their feedback, we refined our reference model. In the second round, we asked four industry experts with experience in mobile application development. In both rounds, each participant received (1) the quality framework including the questions; (2) the mobile checklists' reference model and (3) an Apple iPad with two mobile checklist applications. The checklists were different for each participant and not among the ones used in constructing the reference model. The experiment started with an explanation of the framework given by one of the authors, then the participant had to analyze the reference model with regard to its syntactic quality. Afterwards he or she had to use and analyze the two pre-installed mobile applications. In order to evaluate the semantic quality, we asked the participants to model the checklist application using the reference model. Finally, with a better understanding of the syntax and semantic of the reference model, the participant had to evaluate the comprehension of the model (pragmatic quality). For each participant, the experiment ended with an open discussion in order to clarify the issues raised and gather additional informal feedback.

In the first round, the main comments were related to the model's validity and completeness, more specifically to the entities *participant* and *object.* One researcher suggested moving the subjects at a task level to allow individuals to be involved in one specific task. This comment did not come from the checklist analyzed, but from a personal note. Although it is relevant, we decided not to integrate it in the final model, because we did not observe it in any checklist, and it would imply to assign the participant at the task level. Another comment related to the utility of the *object* entity and suggested merging *object* and *context*. We agree on their similarity in terms of attributes, but from a conceptual point of view, object and context differ. The object is a focal entity for checklists and goes beyond the context: It represents the motivation of an activity (checklist in our case), and once processed, leads to the end of the activity [24]. In the second round, comments on the syntax of the model concerned the entity *checklistInstance.* The latter was first named *instance*, which confused our participants.

Table 2. Quality Criteria for the Conceptual Model's Evaluation (based on [15])

Criteria	Goals	Model Properties	Evaluation
Syntactic quality	Correctness: All statement in the model are according to the syntax.	• Formal syntax	Round 2: • Misunderstanding with the instance
Semantic quality	Validity: All statement made by the model are correct. Completeness: The statement about the domain are correct and relevant.	• Correctness • Annotations and traceability • Consistency • Unambiguity	Round 1: • Utility of having two entities for the context? • Roles at a task level? Round 2: • Roles at a task level?
Pragmatic quality	Comprehension: The model projections have been understood by the relevant audience.	• Executability • Expressive economy • Structuredness	N/A

7 Discussion and Conclusion

As topic of research, mobile checklists lie at the intersection between organizational design and mobile technologies: 1) checklists are cognitive tools that codify procedural knowledge present in organizational routines; 2) mobile technologies can play an important role in supporting individual task execution, thereby maximizing the quality and safety of work. Until now, checklists have been mostly investigated by domain experts who focus on processes and guidelines for creating effective checklists for specific purposes. As underpinned by the calls for a "science of checklists" [2], we lack academic work, which goes beyond checklist design for specific purposes, to advance checklist design, implementation and evaluation. In order to address this research gap, the paper at hand sheds light on checklists' characteristics and conceptualization as well as the specificities of mobile checklists. Main outcomes of our research are a comparative table (Table 1) analyzing paper-based and mobile checklists as well as a reference model (Figure 4), which was systematically constructed following design science guidelines. Coming back to the concept of checklist as cognitive tool, we can argue that mobile checklists provide a better cognitive support for the following reasons: 1) Mobile checklists adapt a checklist's content according to the context, the subject and the object and thereby provide the executants with very specific task documentation; 2) They can automatically verify the consistency of tasks by means of statuses and outcomes; 3) They do not only provide individual guidance for executing tasks, but also help with documentation of the results and collaboration within teams. By providing the vocabulary and symbols for conceptualizing checklists, the reference model creates a shared understanding of the domain knowledge between users, checklist designers and developers. Interestingly, our study reveals two main differences between mobile and paper-based checklists, with impact on the reference model: dynamic adaptation of mobile checklists and the real-time verification of tasks. Since the content of mobile checklists may be dynamically adapted to the context, subject or object, they become more accurate than their paper-based alternatives. In addition, mobile checklists provide real-time verification, by analyzing the status or output of tasks to eventually alert or inform the executants. This functionality, which can at most be realized by a manual verification for paper-based checklists, requires to model constraints in the reference model.

Our research complements existing studies that mainly focus on the usage and benefits of mobile checklists, but do not elaborate on their structure and design. The suggested reference model lays important groundwork by synthesizing the main entities describing the structure, roles and context of mobile checklists and their relationships. It thereby contributes to the science of checklists [2] and advances their development, implementation and evaluation. Practitioners can benefit from our reference model for developing mobile checklist applications faster, thereby decreasing the upfront effort linked to the conceptual design. The model also supports software vendors in offering mobile checklist applications as "packaged software" with configuration possibilities (e.g. similar to Business Process Management tools).

We acknowledge certain limitations of our study. Though our reference model was designed based on scientific literature and the analysis of 22 (mobile) checklists from different fields and evaluated based on 12 additional mobile checklists, we cannot guarantee to have covered all possible conceptualizations of checklists. Furthermore, we only conducted two rounds of analytical evaluation, since we are still at a relatively early stage of reference model construction. This implies that the applicability of the model has not been tested so far, but will be topic of future research. Since the model's main goal is to represent generic domain knowledge, it needs to be included in requirement analysis methods to ensure an effective implementation of a mobile checklist in a specific setting.

To conclude, our research is meant to lay the groundwork for a long-term research that aims to develop a design theory for mobile checklists. Consequently and in order to reach this goal, our future research will investigate the design of individualized and dynamic mobile checklists which leverage the capabilities of mobile devices to effectively support working routines. We see further research opportunities in the experimental evaluation of mobile checklists' usage in organizational contexts and the identification of design principles for effective mobile checklists.

References

1. Hales, B.M., Pronovost, P.J.: The Checklist—A Tool for Error Management and Performance Improvement. J. Crit. Care. 21, pp. 231–235 (2006).
2. Winters, B.D., Gurses, A.P., Lehmann, H., Sexton, J.B., Rampersad, C.J., Pronovost, P.J.: Clinical Review: Checklists - Translating Evidence into Practice. J. Crit. Care. 13, 210-219 (2009).
3. Giessmann, A., Stanoevska-Slabeva, K., de Visser, B.: Mobile Enterprise Applications–Current State and Future Directions. In: 45th Hawaii International Conference on System Science. pp. 1363–1372. IEEE Computer Society (2012).
4. Schmitz, F.M., Zimmermann, P.G., Gaunt, K., Stolze, M., Schär, S.G.: Electronic Rating of Objective Structured Clinical Examinations: Mobile Digital Forms Beat Paper and Pencil Checklists in a Comparative Study. In: 7th Conference of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society. pp. 501–512. Springer (2011).
5. Ockerman, J., Pritchett, A.: A Review and Reappraisal of Task Guidance: Aiding Workers in Procedure Following. Int. J. Cogn. Ergon. 4, 191–212 (2000).
6. Haynes, A.B., Weiser, T.G., Berry, W.R., Lipsitz, S.R., Breizat, A.-H.S., Dellinger, E.P., Herbosa, T., Joseph, S., Kibatala, P.L., Lapitan, M.C.M.: A Surgical Safety Checklist to Reduce Morbidity and Mortality in a Global Population. N. Engl. J. Med. 360, 491–499 (2009).
7. Boorman, D.: Today's Electronic Checklists Reduce Likelihood of Crew Errors and Help Prevent Mishaps. ICAO J. 56, 17–20 (2001).
8. Gawande, A.: The Checklist Manifesto: How to Get Things Right. Metropolitan Books, New York (2010).
9. Elm, W.C., Woods, D.D.: Getting Lost: A Case Study in Interface Design. Proceedings of the Human Factors and Ergonomics Society Annual Meeting. pp. 927–929. Sage Publications, London (1985).
10. Shamo, M.K., Dror, R., Degani, A.: Evaluation of a New Cockpit Device: The Integrated Electronic Information System. In: Human Factors and Ergonomics Society Annual Meeting. pp. 138–142. Sage Publications (1998).
11. Winter, R., Schelp, J.: Reference Modeling and Method Construction: a Design Science Perspective. In: ACM Symposium on Applied Computing. pp. 1561–1562. ACM (2006).

12. Vojislav, B., Leon, J.: Evaluating the Quality of Reference Models. In: 19th International Conference on Conceptual Modeling. pp. 484–498. Springer (2000).
13. Ahlemann, F.: Towards a Conceptual Reference Model for Project Management Information Systems. Int. J. Proj. Manag. 27, 19–30 (2009).
14. Becker, J., Schütte, R.: Handelsinformationssysteme: Domänenorientierte Einführung in die Wirtschaftsinformatik. MI Wirtschaftsbuch (2004).
15. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Softw. 11, 42–49 (1994).
16. Moody, D.L.: Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. Data Knowl. Eng. 55, 243–276 (2005).
17. Moody, D.L., Sindre, G., Brasethvik, T., Sølvberg, A.: Evaluating the Quality of Information Models: Empirical Testing of a Conceptual Model Quality Framework. In: 25th International Conference on Software Engineering. pp. 295–305. IEEE Computer Society (2003).
18. Suter-Crazzolara, C.: Finalists of the "SAP Mobile App Challenge 2012 for Partners in EMEA & DACH," http://scn.sap.com/people/clemens.suter-crazzolara/blog/2012/10/22/announcing-the-three-finalists-of-the-sap-mobile-app-challenge-2012-for-partners-in-emea-dach.
19. Midkif, A.H., Hansman, R.J., Reynolds, T.G.: Air Carrier Flight Operations. MIT International Center for Air Transportation (2004).
20. Goh, D.H.-L., Chua, A., Khoo, D.A., Khoo, E.B.-H., Mak, E.B.-T., Ng, M.W.-M.: A Checklist for Evaluating Open Source Digital Library Software. Online Inf. Rev. 30, 360–379 (2006).
21. De Haes, J.C., Van Knippenberg, F.C., Neijt, J.P.: Measuring Psychological and Physical Distress in Cancer Patients: Structure and Application of the Rotterdam Symptom Checklist. Br. J. Cancer. 62, 1034-1038 (1990).
22. Bézivin, J.: On the Unification Power of Models. Softw. Syst. Model. 4, 171–188 (2005).
23. Chase, D.R., Wegman, M., Zadeck, F.K.: Analysis of Pointers and Structures. In: ACM SIGPLAN 1990 Conference on Programming Language Design and Implementation. pp. 296–310. ACM, New York (1990).
24. Uden, L.: Activity Theory for Designing Mobile Learning. Int. J. Mob. Learn. Organ. 1, 81–102 (2007).
25. Teri, L., Truax, P., Logsdon, R., Uomoto, J., Zarit, S., Vitaliano, P.P.: Assessment of Behavioral Problems in Dementia: the Revised Memory and Behavior Problems Checklist. Psychol. Aging. 7, 622 (1992).
26. Anda, B., Sjøberg, D.I.: Towards an Inspection Technique for Use Case Models. In: 14th International Conference on Software Engineering and Knowledge Engineering. pp. 127–134. ACM, New York, USA (2002).

Table 3. Paper-based and Mobile Checklists Analyzed

		Name	Structure	Task Status	Task Result	Other Elements
Paper-based checklists	Literature	Surgical safety checklist [6]	19 tasks in 3 sections	Binary checkbox	-	-
Paper-based checklists	Literature	Revised memory and behavior problems [25]	24 tasks in 3 sections	Scale from 0 to 4	-	-
Paper-based checklists	Literature	Evaluating digital library software [20]	120 tasks in 12 sections and 42 sub-sections	Binary checkbox	-	-
Paper-based checklists	Literature	Rotterdam symptom checklist [21]	34 in 1 section	Not at all A little Somewhat Very much	-	-
Paper-based checklists	Literature	Detection of defects in use case models [26]	19 tasks in 4 sections	Free text	-	-
Paper-based checklists	Literature	Pre-flight checklist [19]	77 tasks in 3 sections and 5 sub-sections	Binary checkbox	Static predefined outcome	Additional information for each sub-section
Paper-based checklists	Practical-world	Valve installation	25 tasks in 2 sections and 15 sub-sections	Binary checkbox	-	-
Paper-based checklists	Practical-world	How to build a custom home	122 tasks in 20 sections	Binary checkbox	-	-
Paper-based checklists	Practical-world	Crane checklist, construction sites	20 tasks in 5 sections	Various (e.g., yes, partially, no)	-	Pictures
Paper-based checklists	Practical-world	Auto detailing inspection	47 tasks in 2 sections and 12 sub-sections	Binary checkbox	-	-
Paper-based checklists	Practical-world	Patient safety in operating rooms	34 tasks in 3 section and 11 sub-sections	Binary checkbox	-	-
Paper-based checklists	Practical-world	How to organize an IEEE Conference	69 tasks in 5 sections	Binary checkbox		Timeline, links to documents, contacts

		Name	Structure	Task Status	Task Result	Other Elements
Mobile checklists (*Electronic checklist)	Literature	Pre-flight checklist [7] *	11 tasks in two sections	Binary checkbox	Dynamic predefined outcome	N/A
Mobile checklists (*Electronic checklist)	Literature	Electronic Objective Structured Clinical Examination[4]	Tasks in categories and sections	Various (e.g., very good, good sufficient)	Various (e.g., body weight, infection)	Electronic signature, validity constraints
Mobile checklists (*Electronic checklist)	Practical-world	Mobile Service Advisor	40 tasks in 3 sections	Binary checkbox	Various predefined text or free text	3D-models, electronic signature
Mobile checklists (*Electronic checklist)	Practical-world	Clue-in inspection	Customizable tasks in sections	Binary checkbox or selection lists or text	Free text	Pictures, electronic signature, validity constraints
Mobile checklists (*Electronic checklist)	Practical-world	Inspection, checks and audits	Customizable tasks in section and sub-sections	Yes No Na	Free text	Pictures, auditing, reporting
Mobile checklists (*Electronic checklist)	Practical-world	Audit compliance	Checklists built through the app. Sections and sub-sections	Checkbox, free text or selection list	Free text, checkbox, selection list	Pictures
Mobile checklists (*Electronic checklist)	Practical-world	Property inspector	Customizable tasks in section	Binary checkbox	Free text	Pictures, electronic signature
Mobile checklists (*Electronic checklist)	Practical-world	Environmental health inspection	60 tasks in 2 sections and 17 sub-sections	In Out No Na	Dynamic Predefined outcome	Pictures, electronic signature, dashboard, reporting
Mobile checklists (*Electronic checklist)	Practical-world	Mobile field service	Customizable: Built on template	Binary checkbox	Free text in some templates	Pictures, electronic signature
Mobile checklists (*Electronic checklist)	Practical-world	Aircraft checklist (non official)	Customizable: Built on template	Binary checkbox	-	-