

A Web-Based Information System to Evaluate Different Versions of IT Artifacts in Online Experiments

Dennis M. Riehle, Stefan Fleischer and Jörg Becker

University of Münster, European Research Center for
Information Systems, 48149 Münster, Germany
{dennis.riehle|stefan.fleischer|joerg.becker}
@ercis.uni-muenster.de

Abstract. Design science research is a common research method in the field of information systems (IS), as it bridges the gap between IS research and practice. The outcome of design science research are artifacts, which – besides others – can be prototypical applications, so-called IT artifacts. When such IT artifacts are evaluated as part of a design science research cycle, questionnaires and interviews are a commonly used method. We have integrated the demonstration and evaluation phase of a web-based IT artifact, by developing a web-based information system which integrates both a web-based prototype with a questionnaire system. Our information system supports multiple studies, different versions of the IT artifact and several question-specific and participant-specific configuration options.

Keywords: IT Artifact, Online Experiments, Evaluation Platform, Versioning.

1 Introduction

Online experiments are a common research method across a variety of academic disciplines. For example, online questionnaires are used in social sciences to collect empirical data from a large number of participants. In business information systems, questionnaires are, for instance, used to evaluate an IT artifact as part of a design science research (DSR) cycle (e.g., [1, 2]). Consequently, there are numerous application systems available, some also for free and/or as open source products, which enable researchers to create, design and conduct online questionnaires (e.g., LimeSurvey¹). Some tools may even support the statistical analysis of participants' replies.

However, traditional questionnaires are based on a simple question-answer principle. A question is shown to the user on their screen and the answer must be provided as text inputs, options to be chosen or similar methods known from web forms. By this, the question is mostly static, i.e. it consists of text, figures or other non-interactive content. While this is enough for many use cases, like gathering the opinion of users in relation to a certain statement, it may become problematic when evaluating an IT artifact, more precisely, a prototype of a software. In Information Systems (IS), a prototypical software may be the outcome of a design science-oriented research approach,

¹ <https://www.limesurvey.org/>

which bridges the gap between IS research and IS practice [2, 3]. Such research, which nowadays has become quite popular in the IS discipline [4], requires an evaluation from both a user and a technical perspective. While a questionnaire for sure provides insights on the perceived functionality of a user, technical aspects of the software are hard to grasp. This problem increases if the actual functionality starts to differ from the perceived functionality users experience. For example, users may think that they solved a task adequately, while they did not as they failed to recognize all technical functionalities provided by the software.

To enable a comprehensive evaluation of these IT artifacts, the evaluation of the user perspective and the technical perspective need to be integrated. This means that using the software and reporting on the software's functionality should happen at the same time. Traditionally, this would be done by first having the participant use the software, followed by letting the participant fill a paper-based or an online survey. Over the last decade, rapid web application prototyping has gained much attention (e.g., [5, 6]) and not surprisingly, nowadays many researchers build web-based prototypes instead of monolithic applications. Since online surveys are web-based as well due to their nature of being conducted online, in consequence, it is technically feasible to integrate prototypes and online surveys. While this would provide many new opportunities to researchers, as study participants could work with an IT artifact and report on it at the same time, to the best of our knowledge there is no tool on the market yet. Therefore, we ask for the development of an adequate Information System, which enables the evaluation of web-based IT artifacts.

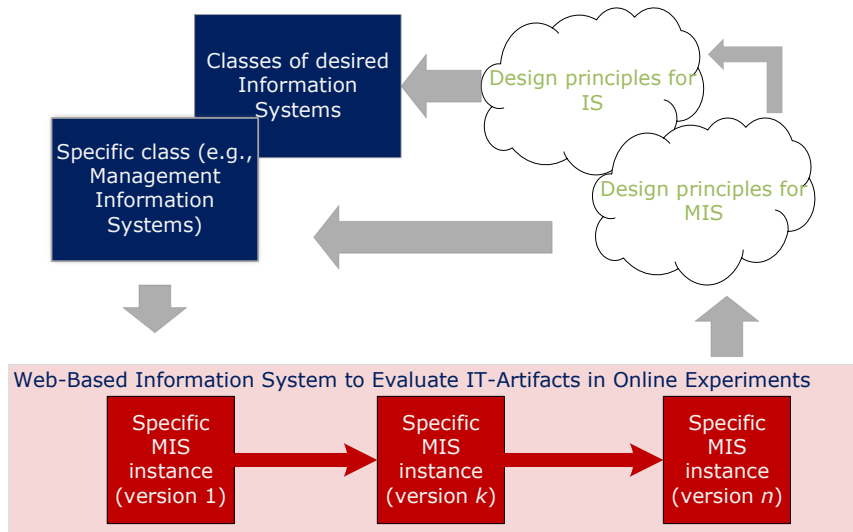


Fig. 1. Research Procedure Model for Evaluating IT Artifacts.

As mentioned, IT artifacts are often the output of DSR. Since most DSR methods are based on a build evaluate pattern, i.e. a repetitive development new versions of the

artifact until all design requirements are met, we expect to have multiple versions of the IT artifact, where each version is the output of a separate design science research cycle. Fig. 1 depicts this procedure. Therefore, an information system to evaluate IT artifacts must allow the integration of multiple IT artifacts, which may be either totally different IT artifacts or different versions of the same artifact to be evaluated. Additionally, many studies require different settings for control groups. For example, if researchers would like to evaluate if a new feature of their software is useful, this feature might need to be enabled for one group of participants but disabled for another group, so that these groups can later be compared. Therefore, an information system to evaluate IT artifacts must on top be able to provide these different settings. We refer to this concept as “disturbances”, as such configuration modifies and, therefore, disturbs individual groups of participants. Obviously, the actual enabling or disabling of the feature must be implemented in the IT artifact itself, but the configuration of this on-off switch should be a part of the evaluation tool.

Therefore, we ask the *research question* of how different versions of an IT artifact can be embedded in web-based online experiments for evaluation purposes. Our *research goal* is to conceptually specify and implement a web-based information system which enables the demonstration and evaluation of different versions of an IT artifact in online experiments.

The remainder of the paper is structured as follows: In section 2 we shortly introduce the design-oriented research approach we follow. Section 3 provides the conceptual specification of our information system, while section 4 briefly mentions our implementation. Section 5 demonstrates our information system in the context of a current research project and section 6 concludes the paper.

2 Research Approach

With our research, we follow the design-oriented information systems research approach by Österle et al [7]. This research approach consists of four steps which are depicted in Fig. 2.

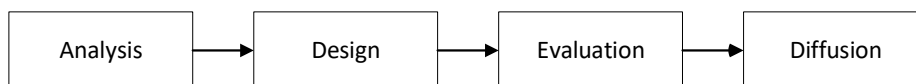


Fig. 2. Design-oriented Information Systems Research Approach [7]

According to Österle et al, design-oriented information systems research is initiated by a stakeholder from scientific or practitioner community or even both. In our case, the evaluation of different versions of IT artifacts in online experiments, the stakeholder is clearly the scientific community. In section 1 we have already identified and described our research problem and specified a research objective. Therefore, section 1 contributes to the *analysis*.

Given our research problem and research objective, in section 3 we *design* an information system by providing a conceptual specification. This specification mostly

consists of Entity-Relationship Models (ERM) [8] and Unified Modeling Language (UML) class diagrams [9]. Section 4 elaborates on our implementation using web-based technologies, which we also use for the *evaluation* of our developed information system, presented in section 5. This evaluation takes part in the context of a research project on intentional forgetting, where we have instantiated our web-based information system. There, we have developed different versions of an IT artifact related to intentional forgetting. We evaluate our information system regarding the effectiveness and usefulness, as suggested by Sonnenberg and vom Brocke [10]. Lastly, the current paper at hand presents the *diffusion* of our work, where we present our findings to the scientific community.

3 Conceptual Specification

Artifacts are a common outcome of design-oriented research (e.g., [1, 2, 7]). While the evaluation of IT artifacts can be performed in many ways, in this paper, we explicitly focus on evaluations in online experiments, since both the technical and the user perspective should be evaluated. For such experiments, users typically are given an IT artifact, use it for a certain purpose and then, after having used the IT artifact, they fill a questionnaire to give feedback. Nowadays, the questionnaire is typically presented on a computer in a web-based scenario, which is why the whole setup is referred to as an online experiment.

3.1 Basic Considerations

The very basic principle of online experiments is the question-answer principle. Participants of the experiment are presented a question page on a screen, where they can fill either one or multiple answers, i.e., a question page can consist of an arbitrary amount of concrete questions. When the participant hits a submit button, their response is stored on the server and, if there are unanswered question pages left, the participant is presented the next question page.

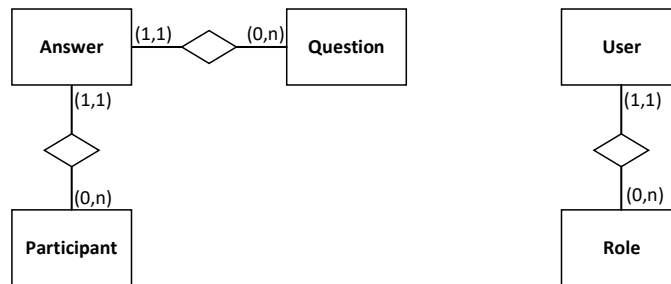


Fig. 3. Basic Question-Answer Principle.

Therefore, for the implementation of our information system, we must consider three different entities in our data model: Question pages, participants and their answers to

these question pages. For simplicity, we are referring to question pages as questions. Given that each question can be answered by many participants, we derive the ERM depicted in Fig. 3. In addition, in Fig. 3 depicts the entities User and Role, where each user is assigned exactly one role. In contrast to the participant, which reflects a participant of the online survey, the user is related to the management area of our information system. Questions can be configured using a web interface which requires authentication. Different roles serve the purpose of authorization. Since the management area of our tool is not in the focus of this publication, we will abstract from the User and Role entity in the further ERMs.

For managing an arbitrary amount of studies in our information system, we need to relate our data to another entity called Study. This entity reflects on the concrete study, e.g., one online experiment for one certain version of our IT artifact. Note that the version of the IT artifact does not directly relate to one study, since multiple studies may need to be conducted for one version of the artifact.

In addition, participants are usually separated into different groups, e.g., a control group and a treatment group. Therefore, we are introducing a new entity called group, to which each participant is assigned. Groups are always assigned to one study, which is what the relationship type between Group and Study reflects. Since there may be questions which only need to be answered by one or more groups, but not by all groups, we convert the relationship between Group and Study to a relational entity type named “Study Group Relation”, to builds another relationship with Question. This enables a configuration of groups per question, where the questions presented during the online experiment can vary per group. In addition, this construct ensures the integrity of our database, as questions and groups are always associated with exactly one study. This clearly separates data from different studies. The resulting ERM is depicted in Fig. 4.

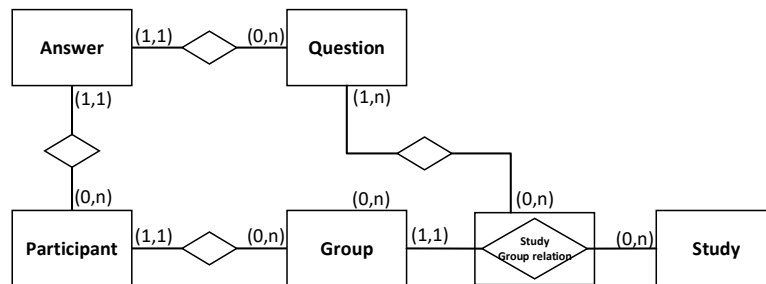


Fig. 4. Data Model for a Multi-Study Tool.

So far, we have recreated a multi-study survey tool. Our information system allows the organization of participants in groups and a per-group configuration of questions presented to the participants.

3.2 The Concept of Disturbances

A common objective of online surveys is testing moderation and mediation variables which are motivated from an underlying theory. In such scenarios, several treatment

groups may exist in a study (a reason to have more than two groups per study), which are supposed to be treated differently throughout the experiment. In terms of evaluating IT artifacts, such variables could, for instance, be the response time or the reliability of the relevant software.

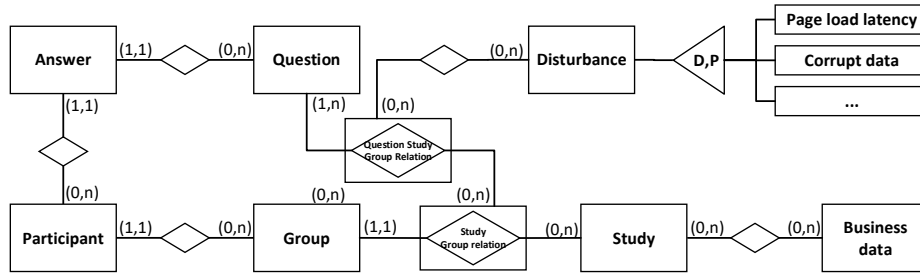


Fig. 5. Data Model with Disturbances.

For our data model, we are referring to such different treatments as disturbances. While a disturbance does not necessarily need to disturb a participant, it changes the appearances of the IT artifact. Obviously, such disturbances need to be configured per group. However, for some experiments these disturbances should only appear for certain selected questions, e.g., only for a few questions, a system failure is simulated. This implies that disturbances need to be configurable on a per-question per-group basis. Therefore, the relationship type between question and the study group relation is again changed into a relational entity type (Fig. 5). This enables to relate the entity Disturbance with the question assignment relation, allowing disturbances to be assigned to certain combinations of groups and questions.

Lastly, we have added a per study generic data storage named Business Data. This entity is supposed to store arbitrary data as JSON strings or key-value pairs. These can later be utilized by the IT artifact. Having such a per study generic data storage allows managing data objects independent of the IT artifact. On the one hand, data objects could be kept over several studies and utilized with different versions of the IT artifact, on the other hand, the same version of one IT artifact can be evaluated with different data objects.

3.3 The Concept of Content Generators

Our information system so far can host multiple online experiments, so-called studies, where participants are organized in groups and must answer an arbitrary amount of questions, depending on the group they belong to. Furthermore, questions can be presented with different disturbances on a per-group basis and we have a generic storage system for data objects.

In the last step, we are now integrating the actual IT artifact into our information system. One first approach might be to assign one version of the IT artifact to the whole study or to assign one version of the IT artifact to each group. However, this does not meet the requirements in practice: In some studies, participants are asked to use two

different versions of the artifact and to compare them. This would require using different versions in different questions. Consequently, we are assigning the version of the artifact to use to the entity Question. In Fig. 6 we refer to the IT artifact as a more generic “content generator”.

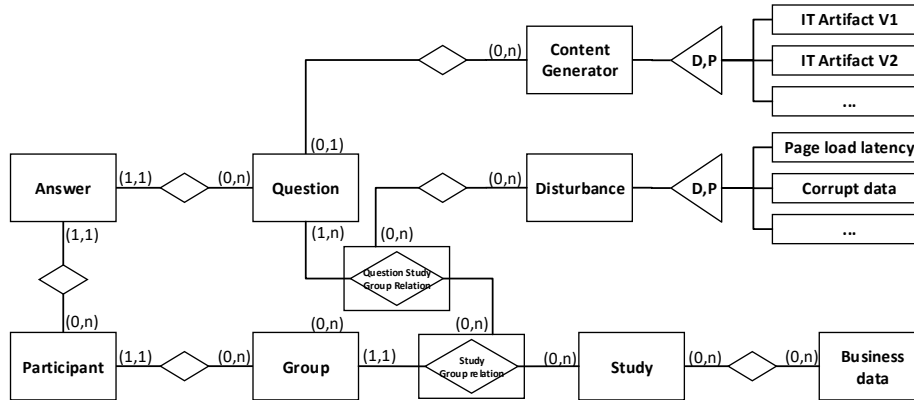


Fig. 6. Data Model with Content Generators.

To integrate the IT artifact in the actual webpage of the question, further considerations are necessary. Since we are focusing on evaluating web-based IT artifacts, i.e., artifacts which are websites themselves, this means that these prototypical applications generate a website based on Hypertext Markup Language (HTML). While further resources, like images, Cascading Style Sheets (CSS) and Javascript files may, of course, be needed as well, the basis of the application is still an HTML string. A content generator, therefore, is an abstraction of the HTML output of the IT artifact, which enables us to embed any kind of web-based application into our information system.

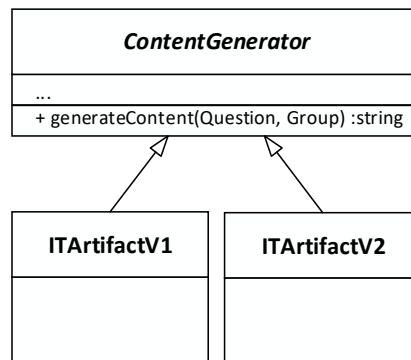


Fig. 7. Class Diagram for Content Generators.

From a technical perspective, a content generator needs to follow a given structure, so that our information system can utilize it. For this purpose, we have designed an

interface, which each concrete content generator needs to implement. Fig. 7 shows the two classes, *ITArtifact1* and *ITArtifact2*, which both implement the interface *ContentGenerator*. This interface only provides one method, *generateContent()*, which is supposed to deliver the required HTML output. The method is passed the actual question and the actual group of a participant so that the content generator may or may not manipulate the output based on the question, the group of the user or the disturbances, which are assigned to this user-group combination.

Whenever a new version of an IT artifact is to be adapted, all that needs to be done is adding a new content generator, which builds the bridge from our information system to the new IT artifact.

4 Implementation Using an MVC Architecture

We have implemented the information system described in section 3 by developing a web application using PHP. PHP is a well know language suitable for web developing which, in contrast to Ruby or Python, has been around for nearly 20 years and is still very actively developed. To better structure our application, we decided to follow a Model-View-Controller (MVC) pattern by using an adequate framework. There is a variety of such frameworks for PHP (e.g., Symfony, Laravel or CakePHP) and for technological reasons of fitting best to our other projects, we have chosen Zend Framework². As a starting point, we used the official Zend Skeleton Application³.

In addition to choosing an MVC framework, we also had to choose a database server to realize the data model presented in section 3. We decided to use MariaDB⁴ since it provides all the features of MySQL but seems to be under more active development (e.g., as of 2018 supports more native JSON features than MySQL). On top of the native PHP support for MySQL/MariaDB, we decided to use a database abstraction layer (DBAL) in combination with an entity-relational mapper (ERM), to encapsulate our data in model objects and separate data handling from our business logic. The most commonly known framework for this purpose is Doctrine ORM⁵. Doctrine provides integration to Zend Framework as well, which supports rapid development, as there are not many things we have had to set up.

```
{
  "name": "institution/mis-evaluation-tool",
  "require": {
    "php": ">=7.2",
    "zendframework/zend-mvc": "^2.5",
    "zendframework/zend-console": "^2.5",
    "zendframework/zend-session": "^2.5",
    "zendframework/zend-navigation": "^2.5",
```

² <https://framework.zend.com/>

³ <https://github.com/zendframework/ZendSkeletonApplication>

⁴ <https://mariadb.org/>

⁵ <https://www.doctrine-project.org/projects/orm.html>


```

        "zendframework/zend-crypt": "^2.5",
        "zendframework/zend-text": "^2.5",
        "zendframework/zend-i18n-resources": "^2.5",
        "zendframework/zend-servicemanager": "^3.3",
        "doctrine/doctrine-orm-module": "^1.1",
        "neilime/zf2-twb-bundle": "^3.0",
        "phpoffice/phpspreadsheet": "^1.2"
    }
}

```

Fig. 8. Exemplary configuration file for Composer.

To manage all external libraries and frameworks, we used the PHP dependency manager Composer⁶. Fig. 8 depicts our Composer configuration file, including all relevant libraries that have been utilized. By using this file, Composer automatically downloads and installs all dependencies. We have added other components – so-called Composer packages – to our application, like the PHP Spreadsheet library, which enables us to generate spreadsheets on the fly. We use this feature to export the collected data in an easy-to-use format for desktop computers.

5 The Case of Intentional Forgetting

To demonstrate and evaluate our information system for evaluating different versions of IT artifacts in online experiments, we have applied our system to one of our current research projects. With our research, we are focusing on the effect of “directed forgetting” known from cognitive sciences [11, 12], which increases the residual influences on forgetting information [13]. Put into an organizational context at workplaces, we are designing IT systems to trigger intentional forgetting. In combination with trust [14], employees may forget information stored by the IT system, allowing them to free cognitive resources they can use for other work tasks.

In our case, we have developed a Decision Support System (DSS), which helps participants in sales planning by providing them with an adequate sales forecast. This DSS is our IT artifact. Over several iterations, we have continuously improved and extended our artifact. The online experiment puts the participants in a fictive organization, the Bicycle Manufacturing Company (BMC). BMC produces different types of bicycles and sport utensils and sells its products in several different countries. To prevent participants from associating sales from country names, only fictive country names were used. The decision task for the study participants is to conduct a sales planning task and maximize the company’s revenue. To do so, participants need to know the historic sales and contribution margins for each product of the last months. Fig. 9 shows the first version of our DSS. The DSS, i.e., the IT artifact, is displayed on the left side of the screen, while the participant’s response is to be entered on the right side. The visual representation of the IT artifact has been retrieved by the evaluation information system

⁶ <https://getcomposer.org/>

via the content generator interface specified in section 3.3. The response boxes on the right side are very similar to a classical survey. Here, the participants enter one number per country, which reflects the number of products the participants plan to produce and sell in that respective country (i.e. a production planning).

Information System (IT Artifact)

Please have a look at the expected sales amount and contribution margins and make a production planning.

Moutainbike, 2017/2018

Country	Sales Forecast	Contribution Margin
Karidol	19	12.50
Laputa	16	7.90
Nautland	13	15.00
Rugiland	14	21.00
Zamonien	20	4.5
	Σ82	Ø12.18
Available Supply: 60		

Your solution (Survey)

Please use the text boxes provided to enter your production planning.

Karidol Pieces

Laputa Pieces

Nautland Pieces

Rugiland Pieces

Zamonien Pieces

Fig. 9. Prototype Version 1.

With feedback and results from the first version of our DSS, we created a second version, which has greater functionality. Here, participants can customize the parameters of the sales forecast and can make different types of forecasts. Fig. 10 shows our updated DSS, where participants can select ratios and dimension from a Data Warehouse. The response boxes are not shown in this figure.

As can be seen, version one and two of our DSS not only differ in styles but also in functionality. Nevertheless, both systems are integrated into our online experiments through the concept of content generators, as presented in section 3.3.

Data Warehouse

Ratio

Date, Department, Place

Cost
 Revenue

[All](#) | [None](#)

Dimensions

Date

Year
 Month

[All](#) | [None](#)

Department

Winter sports
 Bicycles

[All](#) | [None](#)

Place

Country
 City

[All](#) | [None](#)

Fig. 10. Prototype Version 2.

6 Conclusion

In this paper, we have analyzed how IT artifacts can be embedded in online experiments, i.e., surveys, to evaluate their functionality, their usability, their effects on the user or other similar measures. Motivated by design science research methods, where artifacts are created in iterative design cycles, we derived the requirement to evaluate different versions of an IT artifact. We have both conceptually designed and implemented an information system, which serves as a platform to embed not only different versions of IT artifacts, but also to run online experiments with different control and treatment groups.

Our solution is highly customizable due to the configuration, where questions can be individually configured for each participant group and additional parameters (disturbances) can be passed to the IT artifact for further modification. The usefulness of our information system has been demonstrated in the context of developing a DSS which triggers intentional forgetting at the workplace. While the development process of this concrete DSS is not part of this paper, our evaluation platform can easily be used for other kinds of online experiments, where IT artifacts shall be both demonstrated and evaluated.

There are, however, some limitations of our work. With our web-based information system, we have focused on web-based IT artifacts only. Embedding other types of IT artifacts, e.g., Windows or mobile applications is not supported by design. While there are technologies that can bridge the gap between non-web and web applications (for instance, myrtille⁷, a web-based remote desktop client), this has not been the focus of our work. A second limitation is our storage for data objects that we have added to our tool, which only stores abstract data objects. Nevertheless, we think that providing data storages is not the main goal of an evaluation platform like ours. Since IT artifacts usually come with their own technology stack, they will – most likely – have their own data storage anyways. Lastly, while technically any kind of HTML application can be embedded in our information system for evaluation of IT artifacts, an individual content generator must be implemented, which acts as a bridge between the HTML application and our tool. As this means that some little programming effort is required, it is questionable if researchers are willing to do so for their IT artifacts to be compatible. We think that an integration is worth it for conducting online experiments conveniently, but future research still needs to confirm this.

For future work, there are several features that could be added to our information system. Currently, response boxes need to be defined manually by entering an HTML form as part of the question. While this provides maximum flexibility, it is not very user-friendly. Therefore, we might add a form designer, where response boxes can be drawn on a screen. Moreover, participants can currently not be tracked over multiple studies, i.e., participants of different studies are different. This has been a design rationale as well. In cases where the results of a participant need to be compared from one study to another, we might add a functionality, where a participant can participate in more than one study. This would be especially beneficial for long-term studies, for

⁷ <https://github.com/cedrozor/myrtille>

example, when workers are given different types of applications for their daily work and researchers analyze which application best fits the worker's needs. Such experiments usually required measuring the workers' efficiency at multiple points in time on different days, probably even different seasons. However, to provide such functionality in our evaluation tool, we would need participants to have login credentials, which we did not want to have in our experiments.

References

1. Hevner, A.R., March, S.T., Park, J., Ram, Sudha: Design Science in Information Systems Research. *MIS Q.* 28, 75–105 (2004).
2. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* 24, 45–77 (2007). <https://doi.org/10.2753/MIS0742-1222240302>.
3. March, S.T., Smith, G.F.: Design and natural science research on information technology. *Decis. Support Syst.* 15, 251–266 (1995). [https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2).
4. March, S.T., Storey, V.C.: Design science in the information systems discipline: an introduction to the special issue on design science research. *MIS Q.* 32, 725–730 (2008).
5. Nunes, D.A., Schwabe, D.: Rapid Prototyping of Web Applications Combining Domain Specific Languages and Model Driven Design. Presented at the ICWE 2006, Palo Alto, USA (2006).
6. Pop, D.-P., Altar, A.: Designing an MVC Model for Rapid Web Application Development. *Procedia Eng.* 69, 1172–1179 (2014). <https://doi.org/10.1016/j.proeng.2014.03.106>.
7. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum on design-oriented information systems research. *Eur. J. Inf. Syst.* 20, 7–10 (2011).
8. Chen, P.P.-S.: The Entity-Relationship Model – Toward a Unified View of Data. *ACM Trans. Database Syst.* 1, 9–36 (1975).
9. Fowler, M., Scott, K.: *UML Distilled – Applying the Standard Object Modeling Language*. Addison-Wesley, Reading, MA (1997).
10. Sonnenberg, C., vom Brocke, J.: Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In: Peffers, K., Rothenberger, M., and Kuechler, B. (eds.) *Design Science Research in Information Systems. Advances in Theory and Practice*. pp. 381–397. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29863-9_28.
11. Bjork, R.A.: Positive forgetting: The noninterference of items intentionally forgotten. *J. Verbal Learn. Verbal Behav.* 9, 255–268 (1970).
12. Bjork, R.A., Woodward, A.E.: Directed forgetting of individual words in free recall. *J. Exp. Psychol.* 99, 22–27 (1973).
13. Bjork, E.L., Bjork, R.A.: Intentional forgetting can increase, not decrease, residual influences of to-be-forgotten information. *J. Exp. Psychol. Learn. Mem. Cogn.* 29, 524–531 (2003).
14. Thielsch, M.T., Meeßen, S.M., Hertel, G.: Trust and distrust in information systems at the workplace. *PeerJ.* 6:e6483, 1–26 (2018).