A Web Service Architecture for Social Micro-Learning

Bernhard Göschlberger
Research Studios Austria FG
Department of Telecooperation, Johannes Kepler University, Linz
Linz, Austria
goeschlberger@researchstudio.at

Gabriele Anderst-Kotsis
Department of Telecooperation, Johannes Kepler University, Linz
Linz, Austria
gabriele.kotsis@jku.at

ABSTRACT

In knowledge organizations or a knowledge society learning processes are inherently social and distributed. To enable these processes knowledge artifacts need to be created, updated, and consumed decentralized. Social Micro-Learning is an example for an approach following that paradigm. By proposing a flexible service architecture, this paper addresses the diverse demands that Social Micro-Learners have throughout their learning process. It allows integrating information retrieval, recommender systems, workflow engines and spaced repetition algorithms through a single stream data model. Consequently, we can reuse user interface implementations and provide a consistent, recognizable view. Our evaluations show a good system usability, and stable results across different services. We conclude that our service design can serve as a blueprint for social e-learning systems.

CCS CONCEPTS

• **Applied computing** → **E-learning**; *Collaborative learning*; • **Information systems** → *Collaborative and social computing systems and tools.*

KEYWORDS

micro learning, social learning,collaborative computing and tools

ACM Reference Format:

Bernhard Göschlberger and Gabriele Anderst-Kotsis. 2019. A Web Service Architecture for Social Micro-Learning. In *The 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019), December 2–4, 2019, Munich, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366030.3366066

1 INTRODUCTION

Over the past 15 years Micro-Learning evolved from a scientific research area to an immensely popular eLearning business branch. On Web Coursework's eLearning Hype-Curve for 2019 Micro-Learning was one step beyond the peak of high expectations. According to their social media analysis it peaked somewhere around January 2018[1].

Micro-Learning is a paradigm rather than a theory. It postulates a set of requirements, as we will explain in section 3. Most popular Micro-Learning systems today fulfill these requirements and been evaluated in terms of attractiveness and effectiveness (e.g. in [5, 13, 19]) with promising results. The focus of Micro-Learning research is on self-paced individual learning of centrally curated curricular learning content. Thus it has many traits of what Anderson and Dron describe as Cognitive-Behaviourist pedagogy of distance education in [1]. Consequently, we identified Micro-Learning systems focussing on social learning processes as a gap in current Micro-Learning research.

We proposed Social Micro-Learning to enable knowledge sharing rather than knowledge transfer and account for the social dimension of Micro-Learning [12]. It assumes that a central curration of learning content is not possible as knowledge is evolving too quickly. Rather a community of practice (CoP) uses Micro-Content to exchange, discuss and evolve knowledge [20]. Consequently, the Micro-Content units and discussion artifacts become an ever evolving living knowledge repository. The corresponding metaphor—the Knowledge Creation Metaphor— is described in [17]. In the classification of Anderson and Dron learning process would rather follow a Social-Constructivist or even a Connectivist pedagogy of distance education.

In comparison to curriculum based approaches a central problem of evolving content is how to structure and organize it. Depending on the purpose and context many different have to be considered, such as didactic, organizational, or social aspects. In this paper we present our architectural approach to provide a flexible Social Micro-Learning system.

Our research approach follows the design science process for information systems as proposed by [14, 15, 18]. We reflect that in the structure of this paper: following our motivation we show the relevance of our work by situating it in the context of related work in section 2. In section3 we present the design objectives we identified before inferring our solution design in section 4. Subsequently we report our demonstrations in the field where we also evaluated our artifacts in section 5, before concluding in section 6.

2 RELATED WORK

The concept of Micro-Learning was introduced in the early 2000s[9, 10, 16] as a media didactics approach for the demands of a knowledge society. It was envisioned to leverage the potentials of information and communication technology. Micro-Learning is a

[1]https://webcourseworks.com/elearning-predictions-hype-curve/

iiWAS2019, December 2–4, 2019, Munich, Germany

ACM ISBN 978-1-4503-7179-7/19/12…$15.00
https://doi.org/10.1145/3366030.3366066

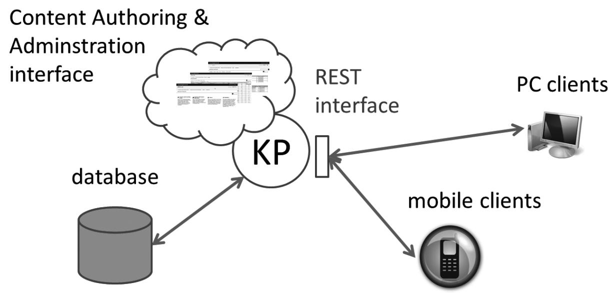

Figure 1: Architectural overview of client-server architecture of *KnowledgePulse* (taken from [5])

pervasive, ubiquitous learning approach. Researchers anticipated the technological advances of mobile and wearable computing devices and aimed to enable learners to integrate learning activities into daily routines and learn during idle times. Micro-Learning connects the intrinsic motivation of learners, that look up information with strategies for long term retention.

Other central pillars of the concept are performance and activity tracking. On the one hand this is necessary for spaced-repetition and mastery learning, which are common adaptive instruction strategies in Micro-Learning. On the other hand there are also didactic considerations. As Glahn points out in his Blog[2], performance is the basis for effective feedback. He subsequently defines Micro-Learning activities as minimal independent feedback loops. Following his argument, we understand a Micro-Content unit as an interactive, digital learning resource that

(1) is self-contained, self-explanatory and can be presented without further context,
(2) comprises a single learning activity that can be performed within seconds, and
(3) provides immediate performance feedback.

Empirical research on Micro-Learning has mainly focused on a certain specific trait of Micro-Learning originally called Integrated Micro-Learning. The conceptual idea of the approach is the delayed access to information services. It seeks to create a small time window for learning before a service is accessed. As an example for the approach Gassler and his colleagues developed a screen saver based application for PCs running Microsoft Windows as a first prototype for empirical evaluations [8, 9].

At the Research Studios Austria FG two further generations of prototypes based on these initial works were developed and evaluated regarding system attractiveness by Bruck et al. [5]. The architecture of the last generation of these prototypes is superficially described as “client-server architecture” as depicted in figure 1. There are no details regarding the data model provided, but the application uses the notion of *course*, *lesson*, and *learning card*. For the latter three different types are described: (1) vocabulary (self-evaluation), (2) multiple-choice single-select, and (3) multiple-choice multi-select.

[2] https://lo-f.at/glahn/2017/06/micro-learning-in-the-workplace-and-how-to-avoid-getting-fooled-by-micro-instructionists.html

Regarding effectiveness this latest generation of these prototypes was evaluated by Smolle [19]. He demonstrated the effectiveness of different micro-content types in terms of short term and long term retention in a pre-test/post-test design.

A similar Micro-Learning system, also focused on the personal rather than on the social learning process is presented by Glahn [10, 11]. He demonstrates the integration and use of standards such as SCORM, IMS QTI and xAPI in the domain of Micro-Learning, and his Mobler Cards app uses three standard compliant Web Services: (1) Authentication Service (OAuth), (2) Question Pool Service (using IMS QTI Information Model), and (3) Experience Tacking Service (xAPI).

In his attempts to situate Micro-Learning in his taxonomy of educational interactions Baumgartner [3] proposes a model of a micro-learner. Baumgartner’s model focuses on informal learning and the learners themselves. He argues that a student has to absorb basic knowledge about a topic or subject in a first step (Learning I), before being able to actively acquire knowledge about that topic in a self-determined manner (Learning II) and finally being able to construct knowledge in a third step (Learning III). With the learner continuing to learn more advanced concepts this process is repeated on a higher level (Learning I+) – leading to an upwards competence spiral. Baumgartner remarks relations between Learning I and behaviorism, Learning II and cognitivism, and Learning III and constructivism. Typical microlearning systems focus especially on the Learning I phase.

Each of those phases demands a different level of guidance and the learning system and peers play different roles. We can define for Learning I, that the software needs to provide defined paths and learning drills. As lined out in the previous section empirical research has focused on this phase and demonstrated the effectiveness and attractiveness of Micro-Learning in this regard. In Learning II phase the learner takes control over his learning process. The system should enable the user to freely navigate through and choose learning resources and the learning systems main focus is recommendation. Learning III phase includes the construction of new knowledge and the system needs to support students to contribute, evaluate and discuss. Social Micro-Learning is designed to better supports students’ development towards and throughout these phases.

3 DESIGN OBJECTIVES

As we follow the Guidelines of Peffers et al. [18] we define the objectives for our solution in this section before we describe the actual design and development of our artifacts. We infer these from the current state of research as described and try to take three perspectives into account as stakeholder perspectives emphasizing the relevance of the objectives: (1) individual learners, (2) communities of practice (3) and organizations

3.1 Design Objectives from the Individual Perspective

First we will list the design objectives for an information system from the viewpoint of an individual learner.

As described above Micro-Learning processes are embedded in everyday life, and the learning episodes are small, informal and relevant. Learners receive immediate performance feedback to learn and improve. Typically Micro-Learning designs for long term retention by using spaced repetition or similar strategies. In Social Micro-Learning we want to use the social context to provide support for Learning II and Learning III. Consequently we try to use it to connect people to learn from each other, and to find relevant content. Social context also serves a source of motivation and empowerment for learners (c.f. [6, 13]). At the same time personal preferences play an important role for learning. On the technological side Micro-Learning needs to be designed for ubiquitous learning, to allow learners to embed their learning activities seamlessly into their daily life.

Consequently we identify the following design objectives:

- Design for MicroLearning
  - DO 1: self-contained and context independent
  - DO 2: single learning activity that can be performed within seconds
  - DO 3: immediate performance feedback
- Design for Long Term Retention
  - DO 4: means for spaced repetition learning
  - DO 5: full control over long term retention learning goals
- Design for Social Context and Social Learning
  - DO 6: personalization for relevance through social context
  - DO 7: enable social learning through exchange with peers
  - DO 8: social feedback mechanism
  - DO 9: express social influence and ownership
- Design for Personalized Learning
  - DO 10: flexible content organization
  - DO 11: integration of external resources
  - DO 12: integration into external resources
  - DO 13: demand driven content search
- Design for Ubiquitous Learning
  - DO 14: mobile-first
  - DO 15: (future) support for multiple clients and client platforms

3.2 Community Perspective

The practices that Communities of Practice (CoPs) cultivate can be very diverse (e.g. music, gaming, science, craftsmanship) and they tend to appropriated available tools for their purposes. It is important to support this appropriation, and defined interfaces to interact with and extend a system are needed. Also meaning and knowledge are emerging within CoPs. New knowledge artifacts become subject of negotiation about their validity and value for the community. Community Information Systems (CIS) need to provide means to discuss, rate and improve items. The different contexts of CoPs impose different requirements on a CIS and it needs to be configurable to support fit multiple contexts. CoPs might as well grow rapidly or share their CIS within a constellation, requiring scalability. But also the structure of a community is dynamic. Thus we can summarize the following design objectives:

- Design for Diverse and Flexible Knowledge Artifacts
  - DO 16: support for diverse representations of learning activities through extensibility
- Design for Emergent Shared Understanding
  - DO 17: support for knowledge artifact centered discussion
  - DO 18: support for collective decision making
  - DO 19: support for retracing and understanding emergence and historic background of knowledge artifacts
  - DO 20: immutability and permanency of artifacts to avoid orphaned references (e.g. in debates)
- Design for Scalability and Configurability
  - DO 21: scalable software architecture
  - DO 22: configurable and scalable content organization
  - DO 23: support for flexible system integration
  - DO 24: support for existing user management and authentication mechanisms
- Design for Community Structures and Evolution
  - DO 25: support to bring together Knowledge artifacts from different sources
  - DO 26: support to use community structure to organize content
  - DO 27: support to use community engagement level to organize content
  - DO 28: support for changes in the community structure (merge, split)

3.3 Organizational Perspective

Organizations are target-oriented and typically competing with other organizations. They have an inherent interest in protecting crucial organizational knowledge that generates competitive advantages. Conversely organizations cooperate with others, share knowledge and form clusters of common interests. These antipodal approaches to knowledge sharing are become apparent in state of the art knowledge management software. By default knowledge artifacts are kept within organizational units, but can be shared if needed. Oftentimes defined processes exist for knowledge artifacts, such as quality assurance or permission granting procedures, requiring defined roles and privileges. We can define the following objectives for the design of the system:

- Design for Knowledge Management
  - DO 29: accountability and attribution
  - DO 30: support for defined business processes
  - DO 31: support for roles and privileges
  - DO 32: support for restricted access and open sharing

4 SERVICE DESIGN

After identifying the design objectives, we present how we designed the solution to address them. We use RESTful web services that are loosely coupled by using the the HATEOAS (Hypertext As The Engine Of Application State) paradim [7] and a configurable web application as a user interface. In the following we will describe our data model and service architecture and their relation to corresponding design objectives.

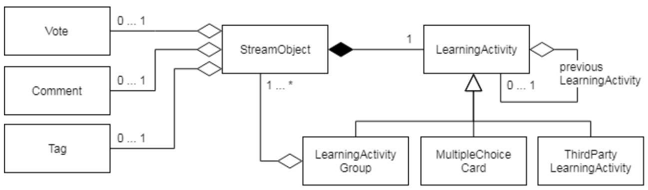

Figure 2: Overview of the Micro-Content Data Model of the Reference Implementation

Design Objectives	Designed Classes
1, 2, 16	`LearningActivity` and its subclasses
7, 8, 17, 18	`Comment`, `Vote`
10	`LearningActivityGroup`, `Tag`
11, 16, 25	`ThirdPartyLearningActivity`
12, 25	`StreamObject`
19, 20	`previousLearningActivity`

Table 1: Design Objectives addressed in the Data Model

4.1 MicroContent Services

The core resource services of the reference implementation are the Micro-Content Services. Their main purpose is to store and retrieve Micro-Content and related information.

Figure 2 shows the data model these core services use. `StreamObject` is the central class, that ties together different aspects resulting from the design objectives. It encapsulates a Micro-Content unit and is addressable through a single resource locator. We chose to call the class comprising of the actual Micro-Content `LearningActivity` to emphasize our paradigmatic focus on interactivity and learning as a product of action, rather than calling it `AbstractMicroContent`.

Since the reference implementation uses the HATEOAS paradigm, client applications can derive all information associated with a `StreamObject` through its resource locator. Similarly, a `LearningActivity` resource provides a resource link to its previous version, if there is any. A `LearningActivity` is immutable (Design Objective 20) and if a user edits learning content a new `LearningActivity` is created. The `StreamObject` always references the most recent `LearningActivity` which references its predecessor. Thus starting from the `StreamObject` all prior versions can be retraced. While a learning activity can only have one predecessor multiple learning activities may evolve from a single predecessor through forking. Forking—as very popular in open source software development—means the creation of a new, independent line of development based on an existing software projects state at a certain point in time. From the moment of the fork two independent strains of development exist. For our reference implementation we adopted this approach, to enable users to fork a `LearningActivity` of another user. This creates a new `StreamObject` with a new `LearningActivity` pointing to the predecessor it was forked from.

Table 1 lists the different design objectives informing the design of the core data model. Each entity has also a `createdBy` property to attribute ownership and support Design Objective 9 and Design Objective 29.

Listing 1: Example JSON Response of a Stream Service

```
{
  "_embedded" : {
    "stream" : [ { ...
      "_links" : {
        "self" : {
          "href" : "<url>/stream/1"
        },
        "learningActivity" : {
          "href" : "<url>/stream/1/learningActivity"
        },
        "tags" : {
          "href" : "<url>/stream/1/tags"
        },
        "comments" : {
          "href" : "<url>/stream/1/comments"
        },
        "votes" : {
          "href" : "<url>/stream/1/votes"
        }
      }
    }, ... ]
  },
  "_links" : {
    "first" : {
      "href" :
      "<url>/stream?page=0&size=12"
    },
    "self" : {
      "href" : "<url>/stream"
    },
    "next" : {
      "href" : "<url>/stream?page=1&size=12"
    },
    "last" : {
      "href" : "<url>/stream?page=2&size=12"
    }, ...
  },
  "page" : {
    "size" : 12,
    "totalElements" : 29,
    "totalPages" : 3,
    "number" : 0
  }
}
```

4.2 Stream Services

The `StreamObject` is the core element of our reference implementation. Its name should indicate that is a single object in a stream of objects. Such a stream of `StreamObjects` is the intended primary delivery unit for the end user. Typically a user interface will deliver the stream of `StreamObjects` using infinite scroll. On the back-end side `StreamObjects` are delivered by a paginated Stream Service.

This is a flexible approach that can be used on mobile devices (Design Objective 14) and other devices of growing importance (e.g. smartwatches, glasses, smart speakers—Design Objective 15). Depending on the client interface, users see a different amount of `StreamObjects` to interact with, and can navigate the content by going forward and backward (scroll, swipe, voice-command). Code Listing 1 shows an example response of a Stream Service illustrating the HATEOAS pattern as well as pagination.

Depending on the context different types of Stream Services delivering different Streams of `StreamObjects` are required. The

Design Objectives	Designed Stream Service
6, 26, 27, 28	Interaction Distance Stream Service
10	Tag Stream Service
13	Search Stream Service
4, 5	Spaced Repetition Learning Stream Service
30, 31, 32	Workflow Stream Service

Table 2: Design Objectives addressed with Stream Services

simplest, straight forward implementation is a stream that is sorted chronologically, descending by creation date (from new to old). While very easy to understand, the usability of this service decreases with growing amounts of activity. However, if we apply content filters, we can control the amount of created content shown in the stream. For our reference implementation we implemented different stream service prototypes to meet different design objectives. Table 2 presents the different design objectives that informed the design and implementation of our stream services. As all stream services provide the same interface and use the same data model, they can be exchanged seamlessly for configuration or even load balancing (Design Objective 21 and Design Objective 22).

In the following subsections we will briefly summarize the design and functionality of our prototypical stream services.

4.2.1 Interaction Distance Stream Service. The interaction distance stream service was designed as a prototype to demonstrate the potential to use interaction data based social networks to filter and prioritize learning activities. It uses an Actor-Artifact-Network (AAN) built from interactions between learners and learning content and uses the elapsed time since the last interaction as a simple edge weight. This network construction is also illustrated in figure 3. The ego-net of each user is than used to display learning content in ascending order of network distance. The hop distance of the ego-net can be configured to adjust the service to network density and size.

4.2.2 Tag Stream Service. The Tag Stream Service provides a Stream of `StreamObjects` that the user tagged with a certain tag. In our reference implementation of a system for Social Micro-Learning users can tag content and use the Tag Stream Service to revisit their tagged content. Thus users can find learning content according to their own categorization and bookmarking. The order of stream items is sorted chronological, descending by creation date. Consequently its implementation is a filtered version of the simple chronological Stream Service.

4.2.3 Search Stream Service. The Search Stream Service returns search results for a user query. The search terms are processed by a search engine—for the reference implementation we used the open source source engine Apache SolR[3]—and the results are returned as a stream of `StreamObjects` ordered by score. Typically users enter a search term in a search bar to optain the search results. Since it uses the same interface as other Stream Services we can reuse the very same views for the search results as we use for the other stream views.

[3]https://lucene.apache.org/solr/

4.2.4 Spaced Repetition Learning Stream Service. The Spaced Repetition Learning Stream Service returns a stream of `StreamObjects` that the user selected for rehearsal. It uses a seperate Learning Analytics Service to retrieve learning records on the bookmarked items. Items are sorted ascending by their last learned date (long ago, before recently rehearsed) and filtered by a spaced repetition logic as described by the pseudocode in listing 2. It is important to note that interaction on the `LearningActivty` is considered for the spaced repetition logic regardless whether it happens through the view presenting the Spaced Repetition Learning Stream or elsewhere (e.g. a Tag Stream View).

4.2.5 Workflow Stream Service. The Workflow Stream Service uses the open source business process engine Flowable[4]. Organizations can model business processes and access rules for the flow of StreamObjects. Each step in the workflow can be used as a Stream Service and configured to be used in the UI with the stream views, since it complies to the Stream Service interface. For instance an organization could model a stage gate quality assurance process, where newly created content is initially only visible for users with privileges to view certain streams (e.g. Review Queue). Once the content is approved, the token containing the `StreamObject` transitions to a different stage of the process (e.g. Internally Available). The modelled processes can be arbitrary complex and the number of different roles is not limited.

5 EVALUATION AND RESULTS

To investigate the impact on usability we did set up three independent evaluation scenarios.

Our first scenario was a workshop on collaborative and social software for learning. Participants were 13 PhD students working in the field of technology enhanced learning. We demonstrated the functionality and people could experiment with the system themselves subsequently. The usability evaluation was conducted qualitatively with semi-structured interviews after the workshop.

The second scenario involved students to prepare for a test in teams using our prototype configured to use the Interaction Distance Stream Service. Each student team had a shared communication channel (slack) to create initial interactions with content of team members. Overall 29 students participated in 4 teams. Participants were co-located during the experiment and they used the prototype for about 3 hours. They were using laptops (8) and smartphones (20), and some of them shared devices.

The third scenario involved students who were asked to create learning material for a course without creating redundant material. This group of students used our prototype configured to use the simple chronological Stream Service. Students were using the prototype remotely over the course of about 3 months. Overall 97 students participated.

For all scenarios the Tag Stream Service was available, while the Search Stream Service was only available for students in the first and third scenario. For the second and third scenario we used the System Usability Score (SUS) to get a quantitative validation of our designed software artifacts [2, 4].

[4]https://www.flowable.org/

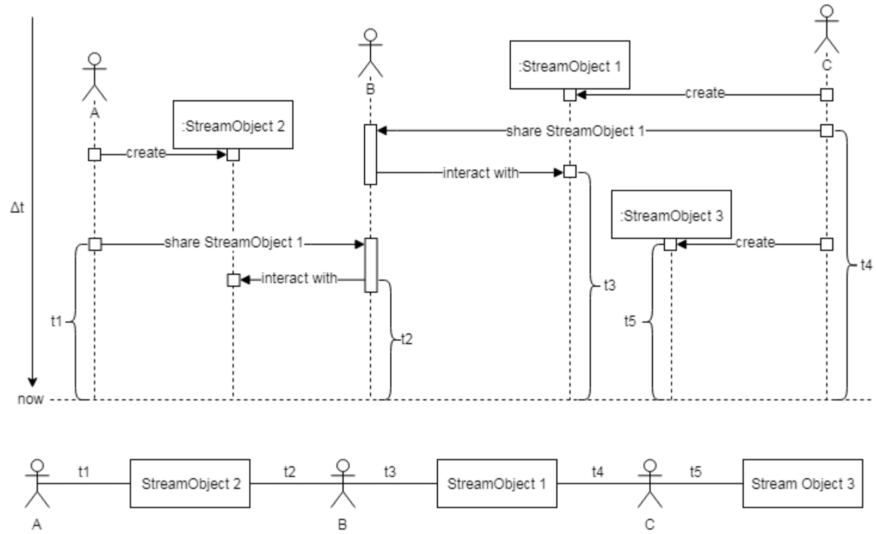

Figure 3: Building an Actor-Artifact-Network based on elapsed time since the last interaction of an actor with a `StreamObject`

Listing 2: Spaced Repetition Logic

```
RepState getRepState(List<LearningRecord> lr, long now){
    long[] spacings = hoursToMillis(1, 6, 24, 72, 168);
    // timestamps of attempts in current success streak
    long[] timings =
        Collections.sort(lr, new NewToOldComp()).stream()
            .takeWhile(r -> r.getResult().getSuccess())
            .mapToLong(r -> parseTime(r.getTimestamp()))
            .toArray();
    if (timings.length == 0){
        // last attempt unsuccessful or not attempted yet
        return RepState.DUE;
    }
    int sIdx = 0;
    int tIdx = timings.length - 1;
    long prevT = timings[tIdx];
    while (--tIdx > 0 && sIdx < spacings.length){
        if (timings[tIdx] >= prevT + spacings[sIdx]){
            prevT = timings[tIdx];
            sIdx++;
        }
    }
    return sIdx > spacings.length ? RepState.DONE :
        now > prevT + spacings[sIdx] ?
        RepState.DUE : RepState.PAUSE;
}
```

Our first evaluation results are based on the responses of five participants that were contacted and interviewed after the workshop. In the semi-structured interviews all participants agreed that the overall usability of the reference implementation is good. Two participants expressed a concern regarding information overload or being unable to find the relevant resources if more people use the system. To reduce the amount of items, different options to filter content for language, topic, or authors were suggested.

Regarding the user experience of the common elements of a StreamObject all participants felt that they were intuitive, and that it was clear what the UI elements were meant to do. A participant mentioned that he felt confident, as the user interface followed standards and widely adopted best practices.

When asked whether they feel to be able to organize their learning content with the reference implementation no participant agreed without restriction. Their restriction was mainly about the type of content they could organize within the system. All participants felt confident that they could use the system to organize their Micro-Content. However, they all felt that it would be hard to convert macro-content to micro-content.

The acceptance and usability of the stream views (in this case: chronological, tagged, and search) was high.

The distribution of the SUS scores of the second and third scenario are shown in figure 4 and 5. The average SUS score was 67.24 in the second evaluation scenario, and 74.25 for the third evaluation scenario. According to [2] the averages for almost 3,500 SUS-Scores in 273 studies is about 70. It is significantly lower for mobile applications with an average of 65.9 and a bit lower for web applications with 68.2. The high amount of mobile users in the second evaluation scenario might have impacted the scores, as well as the shorter usage duration, and the Social Distance Stream Service, which is more complex than the simple chronological Stream Service. Using the statistis of [2], the SUS scores correspond with the adjective *good* and indicate an acceptable system usability.

6 CONCLUSION AND OUTLOOK

In this paper we described the complex set of requirements for Social Micro-Learning that results from the three perspectives of individual learners, communities of practice and organizations. These objectives guided our architectural design as presented. In contrast to other architectures for Micro-Learning focusing on individual learning of predefined and well structured content, we found that certain additional aspects were needed to support social learning processes. These aspects are mainly driven by the needs of communities, and include the need for the system to be extensible and to structure content in different and personalized ways. We

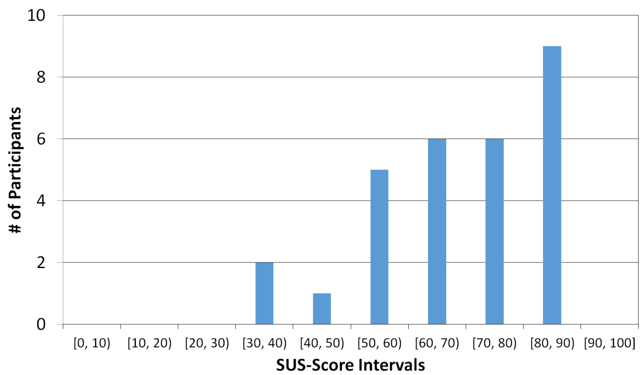

Figure 4: Distribution of SUS Scores in Scenario 2 ($N = 29$)

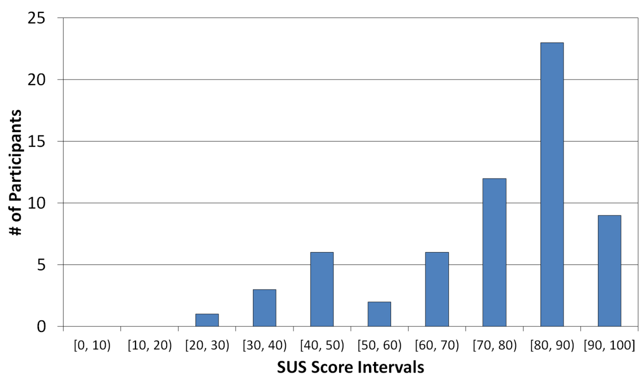

Figure 5: Distribution of SUS Scores in Scenario 3 ($N = 62$)

demonstrated our prototype in practical scenarios, and evaluated its usability. Consequently, we report our findings to the community, before initiating a new design iteration.

We found that our solution works sufficiently well in different scenarios. The implemented Stream Services clearly demonstrate how versatile this approach can be used and the usability evaluation indicates that users understand the system behavior.

In the future we plan to evaluate the different stream service evaluations in depth.

REFERENCES

[1] Terry Anderson and Jon Dron. 2010. Three generations of distance education pedagogy. *The International Review of Research in Open and Distributed Learning* 12, 3 (2010), 80–97.

[2] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction* 24, 6 (2008), 574–594. https://doi.org/10.1080/10447310802205776

[3] Peter Baumgartner. 2013. Educational dimensions of microlearning–towards a taxonomy for microlearning. In *Designing microlearning experiences–building up knowledge in organisations and companies*, Vol. 4. Innsbruck University Press, Innsbruck. http://peter.baumgartner.name/wp-content/uploads/2013/04/Baumgartner_2013_Educational-Dimensions-for-MicroLearning.pdf

[4] John Brooke. 1996. SUS-A quick and dirty usability scale. In *Usability evaluation in industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland (Eds.). Taylor & Francis, 189–194.

[5] Peter A Bruck, Luvai Motiwalla, and Florian Foerster. 2012. Mobile learning with micro-content: a framework and evaluation. *Bled eConference, 25th Bled eConference eDependability: Reliable and Trustworthy eStructures, eProcesses, eOperations and eServices for the Future* (2012), 17–20.

[6] Yu-Kai Chou. 2016. *Actionable gamification: Beyond points, badges, and leaderboards.* https://doi.org/10.1017/CBO9781107415324.004

[7] Roy Thomas Fielding. 2000. *Architectural Styles and the Design of Network-based Software Architectures.* Vol. 54. 162 pages. https://doi.org/10.1.1.91.2433

[8] Gerhard Gassler, Theo Hug, and Christian Glahn. 2004. Integrated Learning with Micro Activities during Access Delays. In *Proceedings of EdMedia + Innovate Learning 2004*, Lorenzo Cantoni and Catherine McLoughlin (Eds.). Association for the Advancement of Computing in Education (AACE), Lugano, Switzerland, 3873–3876. https://www.learntechlib.org/p/12079

[9] Gerhard Gassler, Theo Hug, and Christian Glahn. 2004. Integrated Micro Learning–An outline of the basic method and first results. *Interactive Computer Aided Learning* 4 (2004).

[10] Christian Glahn. 2012. Supporting Learner Mobility in SCORM-Compliant Learning Environments with ISN Mobler Cards. *Connections: The Quarterly Journal* 12, 1 (2012).

[11] Christian Glahn. 2013. Using the adl experience api for mobile learning, sensing, informing, encouraging, orchestrating. In *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2013 Seventh International Conference on.* IEEE, 268–273.

[12] Bernhard Göschlberger. 2016. A platform for social microlearning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 9891 LNCS. 513–516. https://doi.org/10.1007/978-3-319-45153-4{_}52

[13] Bernhard Göschlberger and Peter A. Bruck. 2017. Gamification in mobile and workplace integrated microlearning. In *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services - iiWAS '17.* https://doi.org/10.1145/3151759.3151795

[14] Alan Hevner and Samir Chatterjee. 2010. Design science research in information systems. In *Design research in information systems.* Springer, 9–22.

[15] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. Design science in information systems research. *MIS quarterly* 28, 1 (2004), 75–105.

[16] Theo Hug. 2005. Micro learning and narration: exploring possibilities of utilization of narrations and storytelling for the design of" micro units" and didactical micro-learning arrangements. *Proc. Media in Transition* (2005).

[17] Sami Paavola and Kai Hakkarainen. 2005. The Knowledge Creation Metaphor – An Emergent Epistemological Approach to Learning. *Science & Education* 14, 6 (8 2005), 535–557. https://doi.org/10.1007/s11191-004-5157-0

[18] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2007. A design science research methodology for information systems research. *Journal of Management Information Systems (JMIS)* 24, 3 (2007), 45–78. https://doi.org/10.2753/MIS0742-1222240302

[19] Josef Smolle. 2018. Microlearning an der Med Uni Graz. In *Tagungsband 17. E-Learning Tag der FH-Johanneum*, Jutta Pauschenwein, Helmut Ritschl, and Linda Michelitsch (Eds.). Verlag der FH JOANNEUM Gesellschaft mbH, Graz, 19–25.

[20] Etienne Wenger. 1998. *Communities of Practice: Leaning, Meaning, and Identity.* Cambridge University Press, Cambridge, UK.