N. Gu, S. Watanabe, H. Erhan, M. Hank Haeusler, W. Huang, R. Sosa (eds.), *Rethinking Comprehensive Design: Speculative Counterculture, Proceedings of the 19th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA 2014,* 471–480.

DiNa FRAMEWORK AND PROTOTYPE TO SUPPORT COLLABORATION IN THE WILD

HALIL ERHAN[1], ANDY HUANG[1] and ROBERT F. WOODBURY[1]
[1] Simon Fraser University, Surrey, Canada
{herhan, huang_a, rw}@sfu.ca

Abstract. Much of the available collaboration support tools focus on sharing of documents and managing projects that require planned activities. These tools fall short in meeting principle of least effort or taking into account of the reality of complex work patterns. We propose DiNa framework and system architecture for a topic centric as opposed to document-centric collaboration system using readily available devices. DiNa aims to complement existing approaches. Our primary goal is to seek answers for how these devices can better support collaboration without overloading the workflow. After a literature review and role-playing exercises, the prototypes we developed demonstrate new interaction techniques for defining topics and address them in collaborators' own terms. It uses different visualizations of the artefacts and their association with the topics, among which is a scalable timeline interface accessible from different platforms, to make the artefacts collected more meaningful in a given context. In this paper we present our recent prototype as a proof-of-concept and its initial evaluations followed by the lessons learnt from our studies on supporting collaboration in the wild. The evaluation outcome is suggestions for improving DiNa-based systems for effective collaboration.

Keywords. Collaboration; collaboration support tools; design; mobile computing; distributed cognition.

1. Introduction and Motivation

Effective collaboration not only needs means to generate, share, and manage artefacts to represent products and decisions, but also to trace them as people work on different parts of a task. In a real world, not all artefacts of collaboration are recorded or can be found amongst the others: some get lost or we forget them. Considering the availability of Collaboration Support Tools (CST), why can they not address this issue? The availability answers one part of the problem, while the ‘principle of least effort’ (Zipf, 1949) answers the other. Clark and Gibbs (1986) propose that agents in collaboration minimize their effort in presentation and acceptance of arguments. In general, our entire individual and collective behaviour forms an ecological system in which we attempt to minimize our efforts to achieve a particular goal. If the immediate benefits of using a tool do not outweigh the perceived effort, the tool will not be used. Even small obstacles on the path cause to search for other paths with lesser effort. Therefore we contend that a CST’s success should be measured by how it can address the least effort principle.

In addition to the artefacts, collaboration is based on cues such as body language. If collaborators are thirty meters apart, they might as well be several miles because their interactions are negatively impacted (Allen, 1977). Thus, it is imperative to reduce the perceived distance between collaborators. How can readily available devices reduce this distance? We use them every day to organize our activities and to communicate with others. Can these devices be more support how we work without overloading the workflow?

In this research, we seek answers to these questions. We chose "design" as the domain to experiment with new ideas because it shows characteristics of a complex collaboration environment that can be generalized to others. Previously, we presented how we used role-playing as a means to elicit design requirements for a collaboration tool to complement existing tools (Huang et al., 2011). We also presented a framework informed by the "principle of least effort" and a role-playing exercise (Erhan et al., 2013). The framework proposes a set of recommendations for a topic-centric, as opposed to the document-centric, collaboration system utilizing readily available (mobile) devices. Following these studies, we have developed a series of prototypes with new interaction techniques for collaborators to define and work on *topics* in their own terms. In this paper we present the recent proof-of-concept prototype we built and its initial evaluations followed by the lessons learnt from our studies on supporting collaboration in the wild.

2. Background: On Cognition in Collaboration and Tools

2.1. SOCIAL AND COGNITIVE DIMENSIONS

Hutchins argued that we extend our cognition to external agents to distribute our cognitive load where agents work together by communicating through representations (e.g. drawings) and tools (e.g. computers) (Hutchins, 1995). During collaboration, we capture, process, and share information with others using these tools. These require information to be transformed from one form to another. Collaboration must be simple and clear to make the common context understood by all agents. Kraut, Gergle, and Fussell (2002) showed that having some form of shared visual environment helps improve communication, and drives better results. The shared environment helps collaborators communicate the context and details more effectively.

CSTs should make people feel closer to their collaborators. This is because distance between collaborators matters. Allen (1977) argued that if distance between collaborators gets too far, then the collaboration quality would be negatively affected. While Poster (1995) and Jones (1998) have argued that with the help of digital CSTs, the distance does not matter as much, Bradner, and Mark (2002) have demonstrated that video chat and instant messages alone do not offer enough to bridge the final gap of physical presence. M. Perry et al.'s (2001) interview with mobile workers to understand their collaboration needs. The outcome revealed that rather than highly specific, integrated tools, they prefer lightweight and highly flexible tools. They need their devices to make use of location and available resources as well as taking advantage of connectivity without any other extra task layer. If collaboration tools need extra actions, we may push off the tool use until absolutely necessary or abandon the task entirely.

2.2. EXISTING SYSTEMS AND APPROACHES

Collaboration spans over physical and digital worlds. Peripheral devices and Web-based systems attempt to link these worlds. Web-based CSTs integrate different resources. They are mainly used for four different activities.

Document exchange: The most salient functions of these systems are document storing, retrieving, and exchanging through either an integrated file systems such as EndNote or a virtual file system such as WebDAV. An integrated file system offers synchronous document edits while a virtual file system requires file locking. Collaboration suites such as Microsoft SharePoint and Bentley ProjectWise use both systems.

Provenance tracking: Noel (2005) surveyed nineteen systems, five of which had provenance tracking. Today, most web-based collaboration systems offer some form of document versioning: changes to documents are recorded and reverting to a previous document state may be possible. Blog-like systems allow collaborators to keep a secondary log, similar to journals, of their work in a project. Unlike commit comments in version tracking, this offers the affordance for a more coherent read at review time.

Discussion: There are various types of discussion systems. In the case of Google Docs, discussion is presented as an instant messenger integrated in the document editor. This allows collaborators to chat in an integrated window while working on a document together. In other systems, discussion forums are provided to conduct asynchronous discussions.

Coordination: Coordination is management of dependencies between activities (Malone and Crowston, 1994). It includes, e.g., scheduling, task assignment, resource and project management, conflict resolution, and rationale capture (Medeiros et al, 2001). Web-based collaboration systems for mid-size and larger enterprises usually offer a coordination subsystem.

Although peripheral devices accessing the Web are continuously becoming smarter, collaboration support is still limited. For example, a voice note must first be retrieved from the voice recorder and converted to a commonly workable format. Finally, the record must be shared in a meaningful form. Similarly, sketches require the designer to scan the document first before sharing the information. Smart devices can automate most of these steps and perform actions to enable focusing on the task at hand.

3. Developing DiNa Framework and the Prototype

We developed the DiNa framework and implemented an early proof-of-concept prototype adapting it in three phases (Figure 1). In the first phase, we conducted a role-playing exercise using ‘analog’ props of the mobile devices where the actors were asked to think them beyond how we use them today. We also worked on scenarios based on real-world collaboration cases on West House project (Erhan et al. 2013). The results of the literature review, role-playing exercise and real-world scenarios helped us to outline the high-level requirements for our collaboration framework (Huang et al., 2011). In the second phase, we developed an early prototype adapting the framework. It was intentionally designed and implemented as a low-fidelity prototype to make ‘exploration’ agile. The findings are used to refine and generate ideas for features to iteratively enhance the framework. In the third phase, we evaluated the prototype using a pluralistic evaluation method. In this paper, we revisit the framework with its refinement and focus on the prototype and its evaluation.

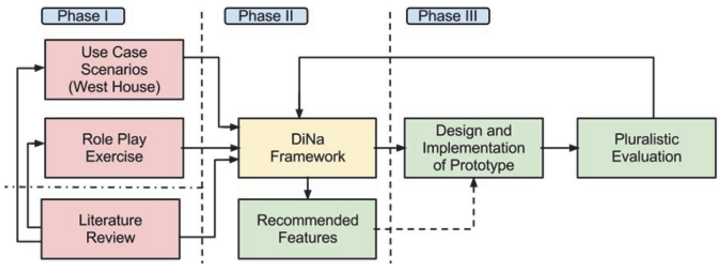

Figure 1. Process of developing DiNA framework and the prototype as proof of concept.

Phase I and Phase II yield to a set of high-level requirements for creating a CST. Relying on these, we developed a system protocol and architecture to facilitate collaboration using any readily available tool.

3.1. DiNa HIGH-LEVEL REQUIREMENTS

By acknowledging the difficulty in compiling comprehensive requirements, we present some of the higher-level system requirements that are essential for effective collaboration.

- **Information capturing and sharing**: Collaborators should be able to record, share, and associate any information when needed. Information can be captured in any form, such as a note, an image, or a video.
- **Access to information**: A demanding function of the current systems is accessing relevant information when needed. A collaboration system should provide features to help user focus on the needed information beyond the platform-specific search mechanisms. Some of them can be access-by-context, direct access, indexed search, or filtering on visualizations.
- **Visibility and control of the dependencies**: The information captured in collaboration forms a complex network. The dependencies in this network can be specific to the project, collaborator, context, or the task. The visualization of these dependencies should make sense to the collaborators through easy-to-navigate visualizations and interactions.
- **Interaction between the collaborators**: Each collaborator in a project has their real-world roles that should not be constrained by system-defined roles. The roles should match the task. In a collaboration system, these roles should be supported with the terms and titles used in the context. Identities and responsibilities should be consistent and their interaction should be simple.
- **Reduce the learning curve**: Most tools require highly specialized training. The tools should be integrated in the larger workflow context by taking advantage of people's experience with existing tools.

3.2. DiNa: LAYERS, ARCHITECTURE, AND PROTOCOLS.

DiNa is an event-centric asynchronous collaboration system framework developed to achieve the higher-level requirements listed above. It uses available technologies to create modular system components with high compatibility and interoperability between them (Figure 2). These components include hardware, such as peripheral devices like mobile phones, smart pens, recorders, and cameras; and software for posting, sharing, retrieving, and editing information captured in different media by different agents.

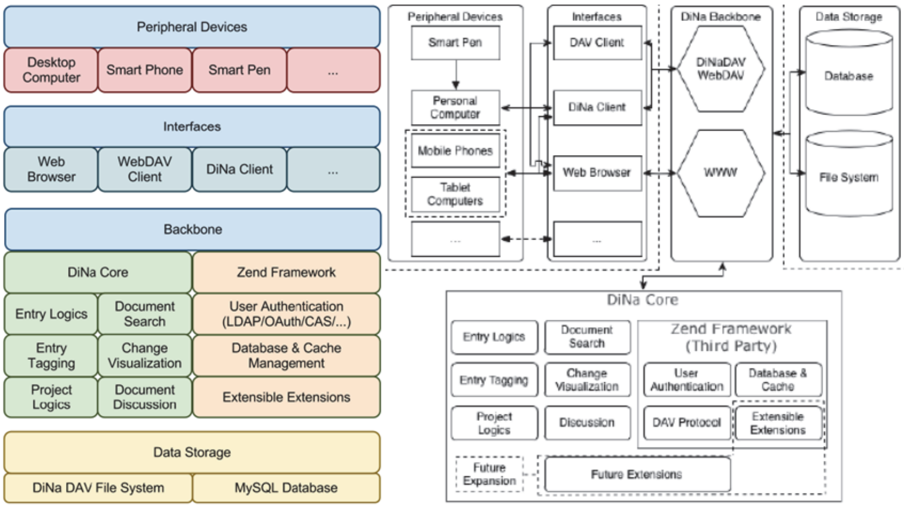

Figure 2. (Left) Layers of DiNa framework and (Right) system architecture showing how the layers are built on pieces of existing technologies and components.

Peripheral devices layer: This layer constitutes `smart' devices with applications for data collection and retrieval. To prevent workflow bottlenecks, they directly access, update, and contribute to DiNa project repositories at the time of interaction. Thus, a series of specially designed applications are needed. Users can perform a task through a context aware application to access the repository with relevant metadata. Non-web-enabled devices (e.g., a smart pen) links with a Web device (e.g. mobile phone) to upload data.

Interface layer: We developed three methods of communication with the DiNa server. The first is the DiNaDAV client, which extends the Web-based Distributed Authoring and Versioning (WebDAV) protocol. The limited implementation enables peripheral devices to access project repositories to open, edit, save, and even delete files from any application. The second uses HTTP to enable most devices with browsers to communicate with the server. Users can setup projects, groups, permissions, milestones, and deadlines. They can also use timeline alongside other filtering techniques such as full text search, date range, tags, location, and other metadata filtering. Full history, in-line discussion tagging, configuring preferences and pushing them across all devices are managed through this interface. The third is device specific interfaces for specialized applications, for example a camera application on Apple iPhone, which must be individually developed.

Backbone layer: The backbone includes server logics into two components. The DiNa Core cluster handles DiNa related logics to manage project, topic and entry logics, search, change visualization, tracking of discussions.

The smaller cluster Zend Framework offers basic of functions such as authentication, database and cache management, and extensibility.

Data storage: DiNa Server is composed of two parts storing data: (a) database where all the meta-data are stored in a normalized relational database, and (b) DAV-based file system handles the low level details of file revisions.

4. Prototype: Implemented Use Cases and User Interfaces

Basic system features implemented in DiNa follow three sets of use cases: project, entry and search use cases. The project use cases are adding, merging, and removing a project. An entry is any kind of artefact to be included as part of a topic. The entry-related uses cases describe how the collaborators add, edit, share, and associate entries. The search in DiNa prototype implements different query methods on entries. The use cases define filtering by tags, data range, participant (role), and entry location. The current prototype implements these use cases. The modularity of the system architecture opens for other future use cases some of which can be device or collaboration-type dependent. We adapted the user interfaces that are native to the device to ensure that collaborators can learn the new interfaces and form new mental models without much cognitive effort. Below we describe some of the main interfaces in relation to how they enable capturing and sharing new artefacts.

4.1. EXTENDED CAMERA INTERFACE

To support quickly toggle to a new topic during discussion, a modification to the native user interface is required. For example, switching to camera in DiNa is not different than using the camera in a mobile device but with direct link to the project data captured from current schedule, location, and the attendees. The buttons can lead to two types of actions: modifying the photo or toggling project topic modes (Figure 3)

Figure 3. Camera application access directly to local and project repository to retrieve relevant context data, which was associated with the picture taken.

4.2. TIMELINE INTERFACE

The timeline in DiNa is the main UI where the participants manage their artefacts. After going through several iterations, the interface supports topic-based visual organization of artefacts on a scalable and semantically zoomable timeline (Figure 4). The interface can be divided by any number of partitions by the user, such as, to define private and public artefacts, artefacts shared by collaborators, sharing artefacts across different projects, commenting or setting priority of artefacts (Figure 5). Each action on an artefact is pushed to the DiNa server and all collaborators are informed of the change.

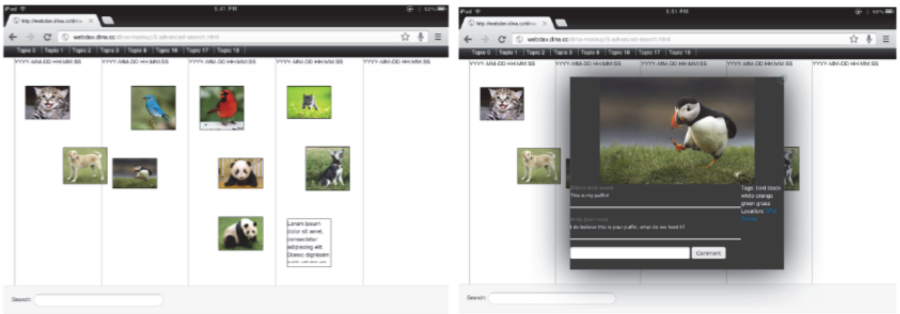

Figure 4. (Right) Timeline interface showing several artefacts shared during collaboration (Right) an artefact is being commented and its priority in topic set on semantic zoom.

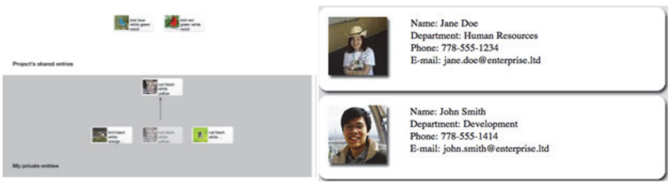

Figure 5. (Left) the topic area divided into different zones by the collaborators to identify related category of concerns (e.g. private and public). The vertical axis defines the strength (e.g. very private or moderately private); (Right) Owner information of an artefact.

5. Evaluation of DiNa: A Pluralistic Walkthrough

We chose to evaluate the DiNa framework and the prototype through a pluralistic evaluation for several reasons: (a) the prototype was partially functional; it was more important to focus on the specific functional details of the framework's recommendations with domain experts; (b) this evaluation gives full control of the prototype to the facilitator, which eliminates the risk of reviewers getting side-tracked by incomplete functions; (c) as a group-based exercise, the participants openly discuss the feedbacks together. These enable participant's feedback to spark conversation with other reviewers, and can reveal more in-depth, focused and relevant findings.

We invited seven reviewers to the evaluation session: four architects and three software designers. Although they were aware of the previous stages of our research, this was their first time to see the prototype. During the session, the main developer of DiNa demonstrated the basic use scenarios. Following a five-minutes formal demonstration on each task, the group discussed the functionality by revisiting the prototype as needed and provided written and oral feedback as to how each task can be further improved. The feedback we received for different use cases is summarized below.

Recording and collecting artefacts: The mobile DiNa client is able to extract project, location, and participant information from the calendar event description. We demonstrated camera feature and showed how it can utilize them. An important feedback was on associating an artefact with multiple topics or partners. The participants suggested using filtering and tagging when we are entering new artefact for creating a multi-topic entry.

Accessing information: Information artefacts stored should be accessed independent from media and location. Upon request, DiNa generates a timeline where the collaborator may review information collected on various topics. The reviewers gave positive feedback on how this is achieved in DiNa, and suggested adding different types of visualization to reveal *deep association* between different artefacts and collaborators.

Searching for information: Search must be available to facilitate locating specific entries in a large repository. We demonstrated the search scenarios DiNa implemented. The reviewers suggested taking further advantage of metadata on other search parameters such as importance or rating on the project timeline. They also suggested interfaces to visualize search on different versions of artefacts as well as a history of search itself. We believe these are highly relevant, but equally interesting features to experiment with.

In each of the task we demonstrated, the reviewers provided positive feedback. Some suggestions from the reviewers also aligned well with the literature. This suggests that the framework has significant potential and encouraging us for further research future development should be done.

5. Conclusions

With complexity, collaboration in the wild is becoming a much more pressing problem. While industry has taken some early steps to make mobile devices work with existing software, these steps are often restricted to only their own system. In this paper, we focused on the problem of ‘How can mobile devices be used for collaboration, and if so, how can we make the process simple?' To achieve this, we proposed a collaboration system framework that may improve collaboration using readily available devices. We evaluated the prototype implementation through a pluralistic evaluation to gain better insights on our approach. We are planning to look into incorporating additional devices and functionalities to better support collaboration. Additionally, we will also look into integrating existing systems to provide more streamlined process. Understanding the ecological validity of our approach requires a comprehensive evaluation of a stable version of the prototype in the wild, which is our next goal.

Acknowledgements

This work was partially supported through the Canadian NSERC CRD with Bentley Systems Incorporated and the MITACS, and the GRAND NCE.

References

Allen, T.J.: 1977, *Managing the flow of technology: technology transfer and the dissemination of technological information within the R&D organization* Cambridge, MIT Press

Boess, S.U.:2008, First steps in role playing. In *CHI '08 on Human factors in computing systems,* 2024-2017, New York, NY, USA

Bradner, E. and Mark, G.: 2002 Why distance matters: effects on cooperation, persuasion and deception. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work,* 226-235, New York, NY, USA

Clark, H. and Gibbs, D.: 1986 Referring as a collaborative process. *Cognition*, 22(1):1-9.

Erhan, H., D. Botta, A. Huang, R.F. Woodbury (2013) Peripheral Tools to Support Collaboration: Probing to Design Collaboration through Role Playing. In the Proceedings of the 18th International CAADRIA, Singapore.

Huang, A., H. Erhan, R.F. Woodbury, and K. Kazlova (2011) Collaboration Workflow Simplified: Reduction of Device Overhead for Integrated Design Collaboration. In the Proceedings of 14th International Conference on CAAD Futures, Belgium.

Hutchins, E.:1995, *Cognition in the wild*. Cambridge, MIT Press.

Jones, S.: 1998, CyberSociety 2.0: Revisiting Computer-mediated Communication and Community. *New Media Cultures*. Sage Publications.

Kraut, R.E., Gergle, D. and Fussell, R.S.: 2002, The use of visual information in shared visual spaces: informing the development of virtual co-presence. In *Proceedings of the 2002 Conference on Computer supported cooperative work*, 31-40, New York, NY, USA.

Malone, T.W. and Crowston, K.:1994: The interdisciplinary study of coordination. ACM Comput. Surv., 26:87-119.

Medeiros, S.P.J; Souza, J.M.; Strauch, J.C.M. and Pinto, G.R.B.: 2001, Coordination aspects in a spatial group decision support collaborative system. In *Proceedings of the 2001 ACM symposium on Applied computing*, 82-186, New York, NY, USA.

Noel, S. and Robert, J.: 2005, How the Web is used to support collaborative writing. *Behaviour & Information Technology*, 22(4):245-262.

Perry, M.; O'hara, K.; Sellen, A.; Brown, B. and Harper, R.: 2001, Dealing with mobility: understanding access anytime, anywhere. ACM Transactions on CHI 8(4), 323-347

Poster, M:1995, The Second Media Age. *Polity*, September 1995.

Zipf, G.K.: 1949, *Human Behaviour and the Principle of Least Effort: an Introduction to Human Ecology*. Addison-Wesley.