Diploma Thesis

Designing mobile and agile groupware for delightful integration into community life

in Partial Fulfillment of the Requirements for the Degree
Diplom-Informatiker (FH)[1]

Presented to the Faculty of Mathematics, Natural Science and Information Science
of the University of Applied Sciences Giessen-Friedberg (Germany) by

Matthias Ansorg

in August 2006

Supervisor: Prof. Dr. rer. nat. Dipl.-Inf. Peter Kneisel
Co-Supervisor: Prof. Dr.-Ing. Klaus Quibeldey-Cirkel

[1]German computer science diploma from a university of applied sciences.

Diplomarbeit

Eine mobile und agile Groupware für das Leben in Gemeinschaften

zur Erlangung des akademischen Grades
Diplom-Informatiker (FH)

vorgelegt dem Fachbereich Mathematik, Naturwissenschaften und Informatik
der Fachhochschule Gießen-Friedberg von

Matthias Ansorg

im August 2006

Referent: Prof. Dr. rer. nat. Dipl.-Inf. Peter Kneisel
Korreferent: Prof. Dr.-Ing. Klaus Quibeldey-Cirkel

Introductory remarks

Version 1.0, published in August 2006

The importance of the glossary. In this thesis, a glossary is employed as a controlled vocabulary. Definitions are centralized within the glossary and appear not in the text. Therefore, it is strongly recommended to read the glossary in full before starting with the first chapter. The glossary terms include some that were coined for the purpose of this thesis. The reason to use the glossary this way is that it enables quite a concise style of writing.

Hints for screen use. This work is optimized for both screen and paper use. It is recommended to use the digital version where applicable. It is a file in Portable Document Format (PDF) with hyperlinks for convenient navigation. All hyperlinks are marked with link flags (▶). Hyperlinks in diagrams might be marked with colored borders instead.

Navigation aid for bibliographic references. Bibliographic references to works which are publicly available as PDF files mention the logical page number and an offset (if non-zero) to calculate the physical page number. For example, to look up [Example :a01, p. 100_{-80}] jump to physical page 20 in your PDF viewer.

Declaration

Official version (German)

Gemäß § 23 Absatz 1 der Prüfungsordnung des Fachbereiches Mathematik, Naturwissenschaften und Informatik der Fachhochschule Gießen-Friedberg für den Diplom-Studiengang Informatik vom 2001-10-24 ▶[FH Giessen-Friedberg :a01, p. 14].

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Translated version (English)

According to section 23 paragraph 1 of the examination rules of the Faculty of Mathematics, Natural Science and Information Science of the University of Applied Sciences Gießen-Friedberg (Germany) for the computer science diploma course of studies from 2001-10-24 ▶[FH Giessen-Friedberg :a01, p. 14].

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Up to now, this thesis was not published or presented to another examinations office in the same or similar shape.

Giessen,

_____

place and date

_____

signature (Matthias Ansorg)

Abstract

Groupware at work and only at work. Much effort has been made both in CSCW research and commercial development to create groupware applications for the workplace. This has been rewarded by the fact that calendaring, task management and project management software is in use within many companies today. Its use can pay off even for small businesses. However, groupware for the workplace has not been adopted in domestic life and community life. Here, a mixture of unintegrated, improvised and idiosyncratic tools is used, both non-digital and digital. Their "frictional loss" when used for group coordination is immense.

The idea of community groupware. The ultimate goal of this thesis is to bring the benefit of coordination support software to private groups, here termed communities. Current societies exhibit a high degree of mobility, so community groupware should work reliably independent of place and other outer circumstances. Life has an often hectic pace of change today, and these changes are often unanticipated. So it is desirable to create an agile coordination tool, which makes community thrive on these conditions and even benefit from them.

Challenges and achievements. There are however several challenges connected with this goal, tackled one by one in this thesis:

1. **Voluntary adoption.** Groupware for the workplace was not adopted in community life yet. To achieve a fundamentally different result, an extensive study of community life is performed before the design process begins.
2. **An adequate organizational paradigm.** Groupware for the workplace is found to be inadequate because of its underlying linear organizational paradigm. This thesis successfully argues that the agile paradigm is a better alternative.
3. **Balancing synchrony and obtrusiveness.** A key problem of telecommunication technology and distributed software systems is the conflict between desirable synchronous communication and desirable

unobtrusiveness. In this thesis, three patterns are proposed to resolve this: "▶Message", "▶Negotiated synchrony" and "▶Subscription".

4. **An adequate communication architecture.** Computer-mediated communication can have totally different qualities from face-to-face communication, including awkward qualities. So extra care is applied to provide a consistent architecture for the style of communication, modeled to resemble face-to-face communication.
5. **Means for agile coordination.** To support coordination in quickly changing environments, this thesis proposes not to implement a static set of features into a CSCW application but instead to design a generic artifact-management application. Both coordinating activity and the current feature set of community groupware will emerge from the behavior of interconnected artifacts.
6. **Communication-integrated coordination.** Socializing communication is a key need of community. To be appropriate, community groupware must respect this. Therefore, this thesis proposes to let coordination be a human task and to support it by facilitated, reliable, unobtrusive and partially automated communication. Communication will consist of messages with a mnemonic text title and audio content, sent and received by mobile phones.
7. **Respecting mobile legacy technology.** To go beyond theory, a prototype of community groupware is developed. This experience leads to increased awareness of the importance to support legacy mobile phones. Community groupware can do so; its messages consist of short text titles and audio content only, and this can be handled by virtually all mobile phones if a legacy interface is provided.

What is better now? This thesis identifies a need, introduces the idea of community groupware to fulfill this need and elaborates on this idea, ending up with a detailed design model called MC³. These are the first three steps to successful use of CSCW in communities. The actual implementation of community groupware is missing yet, but initiated, prepared and simplified by this thesis.

Acknowledgments

Here I really need some space to thank everybody who contributed to this thesis. I want to thank Prof. Dr. Kneisel and Prof. Dr. Quibeldey-Cirkel for their willingness to supervise this thesis. I really enjoyed the large freedom they granted me, especially that I could write on ideas which have been in my mind for some years now.

Further, I want to thank Ortwin Kartmann, manager of Promido Internet GmbH in Butzbach, Germany. He agreed that I could write this external diploma thesis at Promido. It was no external diploma thesis in the traditional sense however; I had large freedom to write on what seemed interesting and challenging to me, and Ortwin Kartmann brought in valuable feedback and ideas in a discussion on the MC³. In particular, I appreciate his ideas for a commercial utilization.

Now I want to thank my parents; you have always supported me during my studies and especially in these last months of writing. I want to thank you, Carolin and Claudia, for proofreading this thesis and correcting all my funny English; I appreciated your help, really. And tank you, Daniel, Joachim and Torsten for valuable feedback or some practical support. Also, thanks to all testers of CGW:RP for their time and willingness to participate and their feedback.

Then, I want to thank all the people who contributed interesting research work which I could quote; and all the people behind the free or publicly accessible Internet and desktop applications which have been such great tools when working on this thesis. Among these applications are: OpenOffice.org, which I would choose again to write a thesis; Google, my favorite search engine; Wikipedia, the free encyclopedia; the LEO German-English dictionary[2]; the dict.cc dictionary[3]; and literally hundreds of other websites and tools, too numerous to write down or memorize. I do not want to forget about these contributions.

To conclude, my personal very special thanks go to three very prominent persons: God the Father, his son Jesus Christ and his Holy Spirit. Thank you for engineering a world where engineering can be so much fun!

[2] http://dict.leo.org/?lang=de&lp=ende

[3] http://www.dict.cc/

Table of contents

Abstract vii

Acknowledgments ix

Table of contents xi

1 Introduction 1
- **1.1** Vision 1
- **1.2** Research space and current focus 4
- **1.3** Current focus in detail 7

2 Background: models about collaboration and community 11
- **2.1** Sociological models for collaboration 14
- **2.2** Abstract organizational models for collaboration 19
  - **2.2.1** Linear organizational paradigm 20
  - **2.2.2** Agile organizational paradigm 23
- **2.3** Concrete organizational models for collaboration 29
- **2.4** Model of community life 31
  - **2.4.1** Organization 32
  - **2.4.2** Preferences 34
  - **2.4.3** Work 37
  - **2.4.4** Communication 39
- **2.5** Model of voluntary technology adoption 43

3 Method: creative engineering, using abstraction and heuristics 47

4 Results: the agile paradigm applied to CSCW 53
- **4.1** Agility: an abstract organizational model for community use 53
- **4.2** MC³: an agile CSCW model for community use 60
  - **4.2.1** Activity set 65
  - **4.2.2** Agile collection 68
  - **4.2.3** Agile message 69
  - **4.2.4** Connectable artifact 77
  - **4.2.5** Conversational message 79
  - **4.2.6** Conversation paradigm 81
  - **4.2.7** Interacting artifact 88
  - **4.2.8** Message 95
  - **4.2.9** Negotiated synchrony 96
  - **4.2.10** Subscription 99
  - **4.2.11** Unbound communicator 100
  - **4.2.12** Universal member 102

5 Implementation: a research prototype 105
- **5.1** Feature set 105
- **5.2** Implementation decisions 106

6 Conclusion: feedback, summary and outlook 111
- **6.1** Discussion of the MC³ 111
- **6.2** Options for commercial utilization 118
- **6.3** Summary of the research 120
- **6.4** Future research 122
- **6.5** Contribution of the research 123

Table of contents

A **Glossary of terms and abbreviations**.....I

B **Source code of CGW:RP**.....XXIII
- **B.1** Instructions and configurations.....XXIII
- **B.2** Class MultimediaMessage.....XXIV
- **B.3** Utility functions.....XXVI
- **B.4** Main program.....XXX

Index of glossary items.....XXXIII

Index of objects.....XXXV

Bibliography.....XXXVII

1 Introduction

Better is the end of a thing than the beginning thereof.

The Bible ▶***[ASV :a01, Ecclesiastes 7:8]***

Summary. The following thesis explores the idea of a groupware which people will appreciate and integrate into their community's life. It starts with a visionary description of the lifestyle that could be enabled by this technology and goes on with a description of the research area and a description of what will be covered in this thesis.

1.1 Vision

Summary. The foundational vision[4] behind community groupware research is presented in three complementary forms: starting with an informal scenario picked out of everyday life, proceeding with a set of claims to be tested, and closing with a short description of a software product based on current technology. When taken together, the three parts should be able to communicate the community groupware vision to the reader.

The vision as scenario. When his smartphone woke him up Padric knew that this was going to become one of those action-packed days. On his phone's display arrived a task to buy some foodstuff; Aleen had entered it at 4:30 o'clock yesterday. She'd been at a wedding party and recognized she wouldn't be up early enough to purchase this as she needed it to prepare the party in the late afternoon. Padric declined the task, proposing to the smartphone groupware to ask another member of his flat-sharing community, preferably Tim. Meeting Enya at breakfast he asked her if he could borrow her car to help a friend move house on Friday; she agreed and Padric used his smartphone to allocate it from the resource pool for Friday afternoon. Padric left at 9:00 for university, meeting some fellow students to do the last preparations for the midterm exams tomorrow. He had agreed to prepare reference material for this group, finishing the job yesterday by doing some 20+ improvements to a previous version; his

[4] Visions as motivation of scientific work bight be unconventional, but they are valid; see Mark Weiser's famous vision of "The Computer for the 21st Century" ▶[Weiser :a01] for another example.

fellows had requested them by sending him subtasks to his smartphone. On his way to university, he got a location-based reminder from his smartphone: Tim had requested yesterday to buy a new bike tire at the bike store; he accepted the task but postponed it as he was quite late. He got the same location based reminder when he left university and got near the bike store; at 17:20 Tim thanked for the tire and the task was marked as "done" and annotated with some accounting information that would automatically balance community expenses at the end of the month. Then Padric helped Aleen to prepare for the party. This was actually a regular event in this flat-sharing community, marking a highlight of the week: inviting lots of friends every Wednesday evening for dinner, talk and some conjoint music performance. Alongside, Padric recorded a message to his smartphone, reminding some of his friends to bring in some new CDs for background music this evening; the note would arrive as multiple-participant task in their smartphone groupware, going through speech recognition before. As Aleen had sent them more tasks than usual to coordinate this mid-week party, he hoped they would still feel just invited and decline them if they considered them a burden.

The vision as claims. Research and engineering may start with ideas and visions, and such is the case here. Ideas and visions are not scientific by themselves but may stand scientific treatment. So let us work scientific now and verbalize the vision as claims that will span the research space for testing them:

- Integrating CSCW into community life enables a different community lifestyle, combining high performance, power, synergy, flexibility and a life without stress in a yet unknown way.
- Adequate CSCW technology for communities has the potential to become a mass phenomenon.
- CSCW technology can decrease the duration of activities by a factor of 2 to 10 by minimizing latency times.
- Using CSCW that way is not only possible in commercial and organizational environments but also in various kinds of communities, as long as the groupware is designed sensitive to this setting.
- CSCW applications will not diffuse into community life until they are excellent, in the sense of "integrable with delight". To get there, engineers

must respect sociological issues and the personal preferences of potential users.

- CSCW applications will not diffuse into community life until they offer a mobile user interface with very high usability.
- All the technical infrastructure needed to deploy such a community groupware is already in place but not at all used to its capabilities yet; it is rather neglected because of engineering difficulties.
- Some research concerning pattern-like verbalization of organizational design and features for community groupware can end up with an adequate and implementation-friendly design model for community groupware.

The vision as product idea. When envisioning technology one should also propose a clue of what is possible to realize in medium-term: a first version of the ready-to-use product. The development of such a product (in this case a community groupware "CGW") must rely on the current state of technology. CGW would exhibit the following technical key characteristics:

- runs on 3G mobile phones
- needs no manual software installation to enable ad-hoc usage, i.e. runs as a web application
- uses an architecture that enables the application to adapt to the diverse capabilities of each of the accessing mobile devices
- is highly optimized for usability, perhaps using an audio-centric user interface and speech recognition to provide convenient input
- is compatible with well-known groupware standards such as SyncML, vCard and iCalendar

The product idea part of this vision complements the scenario and claims and is meant to be equivalent as possible. This should make it fairly clear that this thesis is about application-oriented research, not purely fundamental research. This thesis will contribute some more fundamental research, but only because it is necessary when engineering CGW.

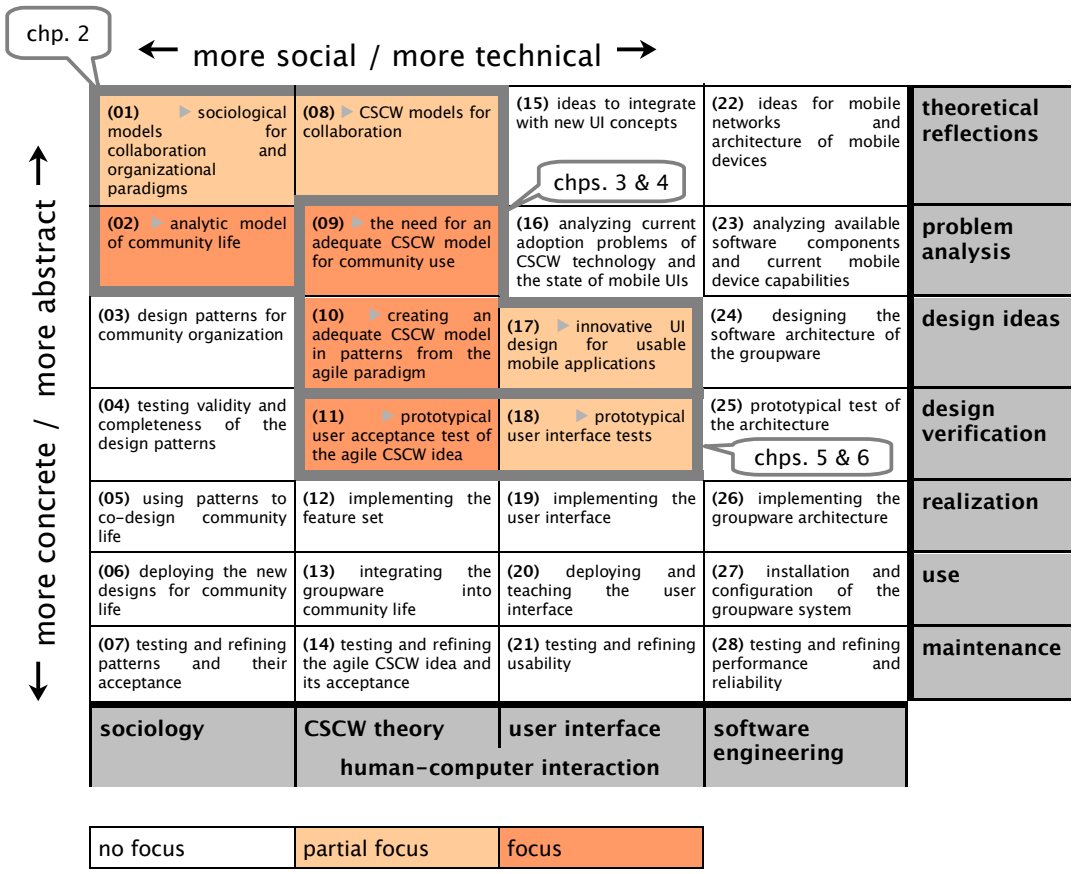
Related research which justifies the vision. Before starting to actually test this vision's claims by scientific work, there should be some reason to assume it

is worth trying. The following points argue that the vision is sensible by pointing to related research efforts which are at different stages:

- A group around Andy Crabtree was occupied with research into ethnographic studies of domestic life and how to support this area with coordination software. See for example ▶[Crabtree :a01], ▶[Crabtree :a12], ▶[Crabtree et al :a02], ▶[Crabtree et al :a04], ▶[Crabtree et al :a07], ▶[Crabtree et al :a11].
- Harry Brignull identified voluntary adoption as a critical point for situated display systems in community settings and placed his 2005 PhD thesis in this area ▶[Brignull :a05]. It seems justifiable to take designing for voluntary (and even delightful) adoption as an important point when designing community software.
- There is remarkable and successful research to find alternative organizational paradigms in quickly changing environment like in modern manufacturing ▶[Paradigm Shift :a01] and software development ▶[Beck :a02], ▶[Beck et al :a03], ▶[Agile Alliance :a01]. This proposes to search for alternative organizational paradigms for CSCW likewise if the current one is inadequate for a new operation environment like community life.
- Team X at NASA's Jet Propulsion Laboratory arrived at high productivity using "warroom" environments, combining social and electronic networking for co-located collaboration. ▶[Mark :a01] coined the term "extreme collaboration" for this.

1.2 Research space and current focus

Summary. The claims of the community groupware vision span a research space, containing a multitude of research tasks to test the claims and transform them to reality where possible. This research space is shown here. It is marked out and advocated which research tasks will be covered by this thesis and which will not be covered.



Object 1: mobile groupware for communities: research space, current focus and mapping to this thesis' structure

See ▶object 1 (p. 5) to get an overview of the research area for mobile groupware applications and the subset which is covered by this thesis▶[5]. Some explanations on what is shown there:

The reasons for covering items and placing the focus. Covering the full research space would mean a full implementation of the first version of CGW. This is beyond this thesis' scope, which will cover only the most relevant aspects of the research space. A review of theories and studies about collaboration and community life is included (items 01–02, 08) as this provides foundations and minimizes redundant efforts. The main difference of CGW to other groupwares

[5]Arguing from the formal requirements for a thesis, the focus is placed such that

- items 01, 02 and 08 review the current state of research
- items 09, 10 and 17 constitute the innovative part
- items 11 and 18 constitute claim tests to verify the proposed innovations

will be its organizational paradigm and its CSCW model which is built upon it (items 09–10) and some innovations on the UI side (item 17). Placing the central research work of this thesis here seems appropriate because human-computer interaction *is* the groupware from the user's point of view; it determines the user experience which in turn determines whether the software will be delightfully integrated into community life or not. This estimation of what is most important for CSCW applications for *voluntary* adoption by communities comes from my personal experiences with two miscarried groupware launches and is confirmed by other researchers:

- John Halloran *et al.* performed an interesting study of a CSCL groupware non-adoption, identifying "a complex interacting set of factors including software use problems, systems integration issues, conflicting tutor/student perceptions of the value of using the groupware, and conflicts in each group's view of how best to complete the course" as the reasons for rejecting the groupware ▶[Halloran et al :a02, p. 169$_{-168}$].
- Matthias Jarke wrote, when he was leading the Fraunhofer Institute for Applied Information Technology (FIT) in 2004: "While the focus was continually on new technologies and features in the last years, now the usability and controllability of these functionalities are rated to be the clincher for market success."[6] ▶[Jarke :a01, p. 44$_{-43}$]. In this article he brings forward the argument that information technology should be designed in human-centered ways and for the acceptance by its users.
- The paper ▶[Halloran et al :a01] is an example for attributing high value to the user experience (including usability concerns, aesthetics and interpersonal aspects) when designing CSCW systems. Here, the risk of non-adoption is met because the newly introduced CSCW tool improves an existing process but does not change its steps.
- Donald Norman widened the user experience concept to include emotions. He successfully argues that attractive things work better because "positive affect enhances creative, breadth-first thinking whereas negative affect focuses cognition, enhancing depth-first processing and minimizing distractions. […] Positive affect makes people more tolerant of minor

[6]The original is written in German.

> difficulties and more flexible and creative in finding solutions." ▶[Norman :a02, p. 36–35].

Finally, a preliminary verification of the claims of the CGW vision and the proposed CGW design is achieved with a research prototype (items 11 and 18).

The reasons for not covering items. Purely sociological research work is a promising add-on for large-scale research work in this area, but as it will not offer practical results in small-scale efforts it is left out (items 03–07 in ▶object 1, p. 5). (Respectively, it is left to research which uses a first version of CGW as a tool for organizational redesign of community life.) While some UI innovations are developed, this is based on CGW's proposed feature set (item 10) and the constraints of mobile hardware and not on a detailed investigation into mobile UI and its usability; items 15 and 16 deal with these and are left out to not distract the focus from organizational design and the CGW feature set. On the software engineering side, CGW will not be fundamentally different from current CSCW applications; so items 22–28 are left out. And finally, the realization, use and maintenance phases of the groupware are excluded. This is partially due to time constraints, and partially due to the hope that a commercial utilization of the CGW idea will result in the longevity of this thesis' results. So the thesis is confined to the theory part mostly (which has some immanent longevity …) and items 05–07, 12–14, 19–21, 26–28 are left out.

1.3 Current focus in detail

Summary. The research space and its coverage in this thesis was already shown in ▶object 1 (p. 5). Here, the covered research items are cited from ▶object 1 (p. 5), including their numbers, and their meaning is explained in detail.

(01) sociological models and organizational paradigms for collaboration

> An overview of research work that can be utilized when dealing with collaboration. Five sociological theories are referenced (see ▶object 4, p. 14) and two conflictive organizational paradigms; where meaningful, valuations and connections are made. Central publications are: ▶[Alterman et al :a10], ▶[Beck :a02], ▶[Crabtree :a01], ▶[Crabtree et al :a02], ▶[Crabtree :a12], ▶[Pankoke-Babatz :a01] (besides other work of these authors).

(02) analytic model of community life

This contains information to understand what community is and how it works. It is derived from extensive study of research work in this area and from some personal observations. Four areas are covered: communities' organization, preferences, work and communication.

(08) CSCW models for collaboration

Some short insights into current and upcoming models to structure CSCW applications, drawing from publications such as ▶[Pankoke-Babatz :a01], ▶[Xu :a01], ▶[Frank :a01], ▶[Mark :a02], ▶[Alterman et al :a02]. This extends the previously treated models of social behavior and the organizational paradigms towards application in CSCW software.

(09) the need for an adequate CSCW model for community use

The current organizational paradigm of CSCW applications is evaluated for correlations with the community's underlying organizational paradigm. The correlation turns out to be very low: current groupwares are unapt for supporting communities in their everyday activities. This is not just a lack of usability; it is argued that community groupware needs a new and adequate organizational paradigm. The agile paradigm is found to be such an alternative.

(10) creating an adequate CSCW model in patterns from the agile paradigm

This is the key innovation within this thesis. The agile paradigm as applied in the manufacturing industry and for software development is applied to CSCW. The goal is to minimize the overhead of groupware usage and end up with a helpful CSCW tool for community's everyday activities. The result is a feature set for CGW, presented as interaction patterns.

(11) prototypical user acceptance test of the agile CSCW idea

A limited subset of the proposed CGW features is implemented into a research prototype. The prototype is used in a community to evaluate the design.

(17) innovative UI design for usable mobile applications

Mobile devices suffer from limited usability, and CGW will run on mobile phones which exhibit the worst restrictions. The proposed features of CGW

are developed bearing these limitations in mind; the result is an innovative UI design, relying mostly on people's most intuitive interfacing method: their voice.

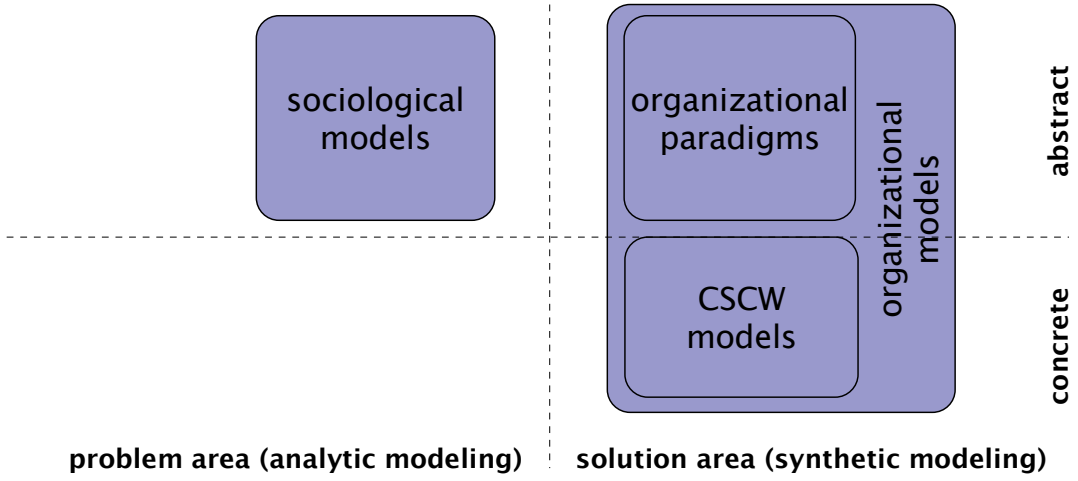
(18) prototypical user interface tests

The user interface tests are integrated with the usage of the research prototype CGW:RP in a community. This prototype implements the key ideas but no fine-tuned UI; the tests are therefore not usability tests but evaluations of the innovative UI ideas as such.

2 Background: models about collaboration and community

> *The design of community networks can support positive values in this complicated world, but only so long as the designers understand what they are getting into.*
>
> ***Phil Agre ▶[Agre :a01]***

Summary. Collaboration models from three different categories are reported on: sociological models, organizational paradigms and CSCW models; ▶object 2 (p. 11) contextualizes them. To communicate them efficiently, a controlled vocabulary is employed. The discussion of CSCW models will later on help to identify the shortcomings of current CSCW applications in community use. As another main point, many insights into community life are collected through extensive literature study.



Object 2: different categories of collaborational models in context of each other

The controlled vocabulary. To present models and theories of collaboration in a concise and unambiguous manner, a controlled vocabulary is used. Its didactic intent is to enable precise and quick access to the topic. The controlled vocabulary consists of terms defined in the glossary (▶pp. I).

It might be a rewarding task to develop a meta-model of collaboration, introducing much more clarity in communicating about it — a good starting point

is Xu's taxonomy of communityware and groupware ▶[Xu :a01] and Crowston's taxonomy of coordination ▶[Crowston :a02]. This however is well beyond this thesis' scope. So this thesis has to be content with a controlled vocabulary as a tool limited to its own scope.

What "model" means here. The theories on collaboration to be discussed are, from a system analysis point of view, models. Here, the term "model" is employed in a loose manner, without any syntactical restrictions on the model's format implied. In this sense, a model is a collection of free-form statements which rely on the controlled vocabulary. They might be in natural language, pattern format or anything else. The statements might be clearly separated and given an identifier, but there is no need to do so.

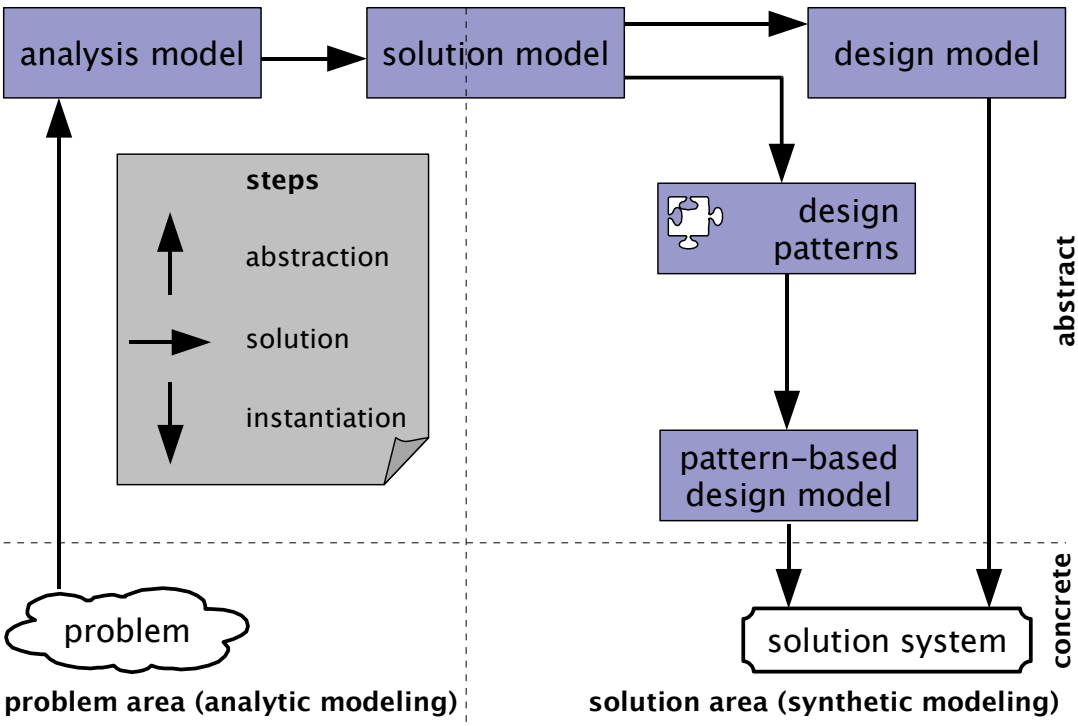
On models in social sciences. In the following subsections, models of collaboration and community which have already been developed by other researchers will be reviewed. This will not end up with "the only complete and true model" however — our fuzzy perception of the complex connections in social domains (or the fuzzy nature of these domains themselves, who knows?) does not allow such a claim. As usual in social sciences, different models are presented, each highlighting specific aspects and complementinge the other ones. Instead of a unification, hints will be provided how to connect them and to translate between them where applicable. The models will help to understand CSCW better even without a unified model.

Relating models to patterns. There is an analysis model for every problem; and perhaps a (hybrid analytic/synthetic) solution model. The first covers the full domain of the problem and explains why the problem is there. The latter is a theory that solves every problem of that kind. Now, learning such a theory and solving problems with it requires effort. And that is where patterns come in: patterns serve as sharable pre-calculated solutions. An additional reason to use patterns is to enable problem solving even where only heuristic[7] solutions but no solution model is available.

In most practical cases, the solution model is incomplete. So it must be supplemented with a pattern collection, providing heuristic solutions. This will be

[7] or accidental, unteachable ingenious or unteachable experience-based

the case in this thesis when proposing the agile paradigm as an alternative organizational paradigm for CSCW.



Object 3: placement of different model types in the system engineering process

Here, analysis models, solution models, patterns and design models are all considered artifacts of systems engineering which differ in their level of abstraction[8] (cf. ▶object 3, p. 13). They do not differ from the syntactical point of view, however, as they may share the same meta-model. Choosing a proper meta-model is therefore a means to integrate newly developed patterns with the work of other researchers.

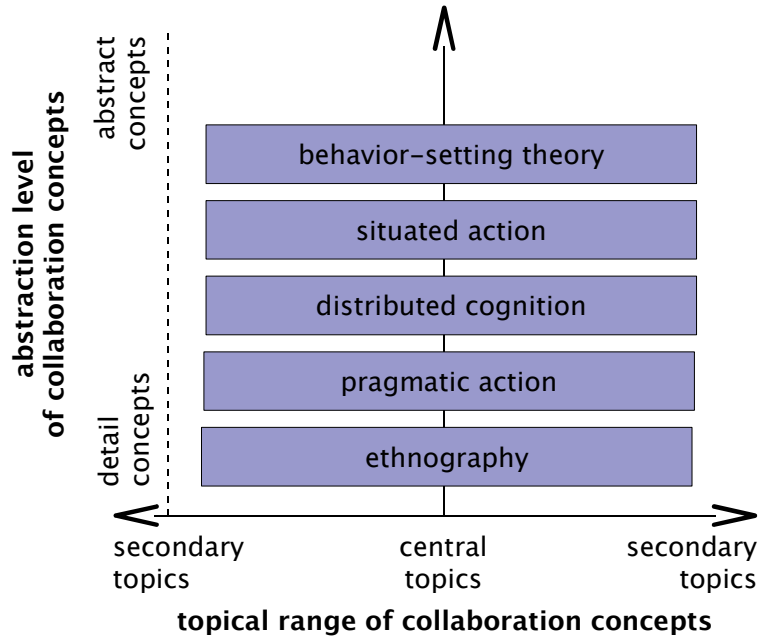
PLML as interaction pattern format. No restrictions have been imposed on the format of the presented models yet. For writing patterns however, a more formal format will be advantageous. While the pattern community basically agrees on what patterns are, the format for presenting patterns varies according to the pattern's purposes. For patterns in interaction design for example, ▶[Fincher

[8]In this view patterns are only instantiated in synthetic models. The existence of analysis patterns does not contradict this: when analyzing the analysis process, each analytic model turns out to be a synthetic one at the same time. Because analysis is about building an intentionally simplified model of reality, choosing from infinite alternatives of simplification. Analysis patterns solve the recurrent problems of simplifying reality.

:a01] gives an overview of the different formats in use. The same author contributed to a major proposal for a pattern format for sociotechnical patterns, called PLML ▶[Fincher :a02]. ▶[Schümmer :a03] provides some more details on PLML and discusses its usage in a system for evolving pattern languages. ▶[van der Veer :a01] goes in the same direction, discussing how to structure a pattern language for human-computer interaction. As PLML is well-fitting for the sociotechnical patterns to be developed here, it is employed as the pattern format in this thesis. The patterns are proposals however, so it will be agreeable to present them in concise manner, leaving out metadata fields and those with self-evident information. The patterns' bibliography is not shown in the "literature" field of the PLML pattern format but integrated with the bibliography of this thesis.

2.1 Sociological models for collaboration

Summary. Choosing from models that are traditionally connected with CSCW, those which are applicable to home and community life are presented. The selected theories are ordered by their scale: the theory with microscopic view comes first, the theory with macroscopic view comes last.



Object 4: domain coverage of referenced sociological theories (navigational aid)

Ethnography in design. For several years now there has been a strong connection between ethnographic studies and the design of CSCW systems; that is presented in detail in Andy Crabtree's book ▶[Crabtree :a13] and an overview is given in ▶[Crabtree :a12, pp. 2–3]. Ethnography was employed for CSCW design in search for solutions to the requirements problem. CSCW tools are supposed to support current organizational practice, but prior approaches for studying the design space have not been able to provide adequate representations of work as carried out in reality. For example, human-computer interaction focuses on individual cognition only, ignoring the actual work to be supported by CSCW systems. Confer ▶[Crabtree :a13, p. 3].

While it is comparatively easy to define functional software requirements for business workflows and other processes of highly ordered domains, it is difficult for tools which support unstructured collaboration. Here, "supporting" concrete qualities of organization cannot refer to collaboration steps or the like but to upper-level concepts like the "character of collaboration", "collaboration style" etc.. Those concepts cannot be transformed to functional requirements and software features in a quasi-mechanical way, they are difficult to recognize and to formalize as requirements.

To cope with this problem, CSCW design has adopted ethnography. It provides informal in-detail descriptions of real-life activities to sensitize system designers to what is going on there. It is important to see that ethnographic data is neither a specification nor a theory: it is material to reflect on, but in itself it specifies or explains nothing. It is merely a tool, e.g. employed as central in the distributed cognition framework ▶[Rogers :a02, p. 732$_{-730}$], ▶[Crabtree :a12, pp. 2–3]. Ethnographic data is not a list of functional requirements, as those are not derivable from unstructured work practices. It wants to sensitize system designers, i.e. to be a tool for estimating creative inventions of features by reasoning about their effects.

Collaboration as pragmatic action. One should not expect sociological models to reveal concept-laden analytic algorithms in the psychology of the individual which underly collaboration. The opposite seems to be true: the psychology of the individual supports not analytic but pragmatic action natively ▶[Alterman et al :a10]. Therefore pragmatic action is most native for the

everyday task environment[9]. The tasks in this environment have no prescribed way of solution attached. They might include collaborative tasks with a likewise unrestricted way of solution; which makes unrestricted collaboration a form of pragmatic action.

Connecting pragmatic action and patterns. There is some interesting research work on patterns of social interaction and daily routine, for example ▶[Crabtree :a01], ▶[Crabtree :a08] and ▶[CPSR :a01]. In the view of collaboration as pragmatic action (see above), these patterns are verbalizations of the skill acquired by pragmatic action. This is because pragmatic action is defined to include skill acquisition through accumulation of experiences within a task environment:

> Because of the semi-permanence of home task environments, there is pay-off in organizing behavior in terms of the particulars of those environments. ▶[Alterman et al :a10, p. 53$_{-52}$]

> In our view, skill acquisition is a phenomenon arising from several sources. [It] […] can result from the mental transformations of knowledge such as proceduralization and chunking; […] from the accumulation of local optimizations to an activity. Our work shows that similar performance improvements can be achieved through the acquisition of task environment specific facts that improve the fit between the actor and his world. ▶[Alterman et al :a10, p. 93$_{-52}$]

If patterns are skill verbalizations, then skill acquisition in the everyday task environment is pattern development in practical life: developing solutions to recurring problems in contexts. Pragmatic action thus is not structure-less but leads to successful patterns of action. As the everyday task environment is only semi-permanent, these patterns change and improve constantly.

Distributed cognition. A framework for conceptualizing cognition, developed by Hutchins *et al.* from 1995 on, introduced in ▶[Hutchins :a01]. Yvonne Rogers defines distributed cognition in an encyclopedia article as follows:

> Distributed cognition is a theoretical approach that is concerned with the interactions between people, artifacts and both internal and external representations. Rather than focusing exclusively on an individual's internal cognitive processes, that traditional cognitive approaches do, it focuses on the processes that take place in an extended 'cognitive system'. These include verbal and non-verbal behavior, the coordinating mechanisms used by social actors, the forms of communication that take

[9]It is a good question, deserving further investigation, if this is "programmed" by the everyday task environment (favoring pragmatic solutions) and / or determined by the neural network structure of the brain.

> place and the way tacit and explicit knowledge is shared and accessed. One major benefit is the explication of the complex interdependencies between people, artifacts and technological systems that can be often overlooked when using traditional theories of cognition. ▶[Rogers :a02, p. 731$_{-730}$]

Distributed cognition is similar to the "pragmatic action" approach of ▶[Alterman et al :a10] (mentioned above), because both deal with information that is encoded into artifacts. While the "pragmatic action" approach describes in detail how this information is used by individuals, "distributed cognition" heads for a wider framework of modeling activity; but not as wide as the behavior-setting theory, which tries to give a closured model of behavior factors, thereby looking nearly completely away from the role and contribution of *single* artifacts. There are examples how distributed cognition can be utilized in the CSCW domain, e.g. ▶[Brignull et al :a01].

Situated action. This perspective on social interaction was introduced by Lucy Suchman in ▶[Suchman :a01]. It has become one of the most widely used CSCW perspectives on human action ▶[Dourish :a02, p. 466$_{-464}$] and can be defined as follows:

> Drawing on ethnomethodological foundations, the situated action perspective looks on the sequential organisation of action as a moment-by-moment improvised affair, emerging in response to the circumstances of its production – physical circumstances, social circumstances, organisational circumstances and so forth. ▶[Dourish :a02, p. 466$_{-464}$]

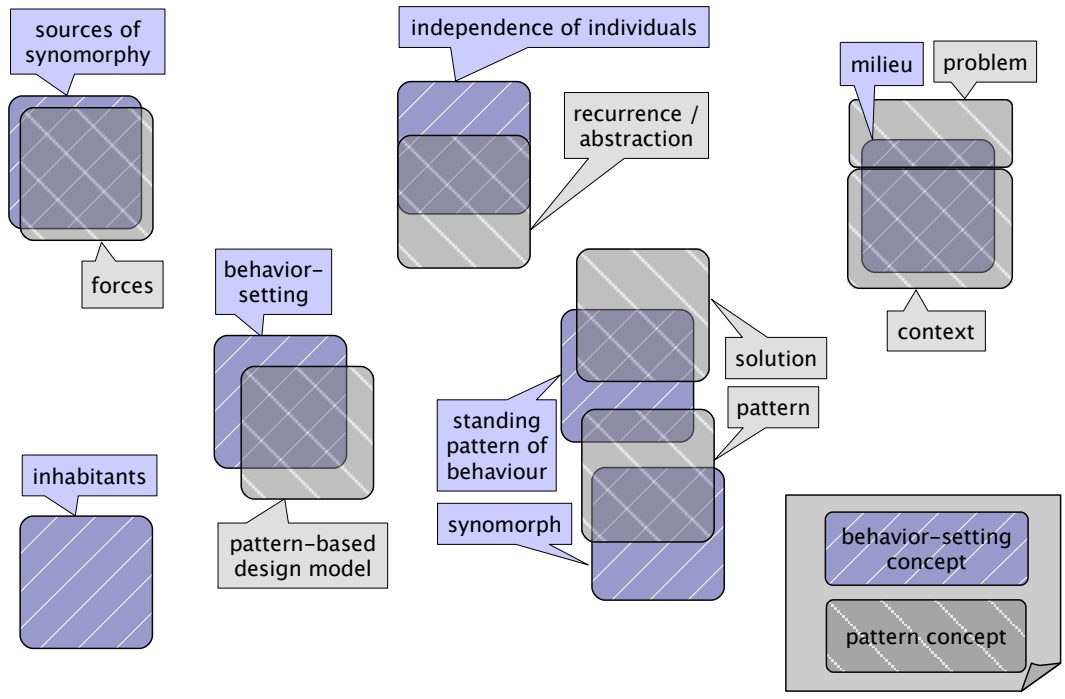
Situated action was chosen by Harry Brignull to study voluntary adoption of community displays in his doctoral thesis. There, he states the following reasons:

> This perspective enables researchers to address questions about how users react to a system when they use it for the first time, how they then learn about the system and its features, how adoption takes place socially between immediate colleagues or friends, and what social roles it develops through use. ▶[Brignull :a05, p. 19$_{+3}$]

Situated action and the behavior-setting theory can be connected as follows: both deal with the influences of the locational and situational context, but while the behavior-setting theory looks on already fixed patterns of behavior that emerged in stable environments, the situated action view is concerned with the structure-less improvised nature of action where no patterns have been developed yet.

The behavior-setting theory. This theory was developed by Barker in his 1968 work "Ecological psychology" ▶[Barker :a01] and deals with the effects of environmental aspects on human communication and (co)operation. Uta Pankoke-Babatz brought it to use as a tool for CSCW design in her doctoral thesis ▶[Pankoke-Babatz :a01]. For further information see her detailed presentation of this theory ▶[Pankoke-Babatz :a01. pp. 19-50$_{+16}$] including its complementation with other sociological theories around communication and action ▶[Pankoke-Babatz :a01. pp. 51-94$_{+16}$].

The behavior-setting theory has its perspective restricted to what is observable; this perspective simplifies the development of a theory. It takes not into account any intentions. The key finding of the behavior-setting theory is a statistically significant correlation between places and behavior. The theory simply is not about deviant behavior, which is seen as statistically exceptional. Despite this limited character, the theory seems very relevant for developing community support tools: it is about places that provoke behavior, so it will help to design the CSCW application as a "digital place" which provokes delightful integration into community life as the "standing pattern of behavior".



Object 5: correlation of behavior-setting and pattern concepts

Connecting behavior-settings and patterns. The behavior-setting theory sees at least some behavior as dependent on the setting, not the individual. Likewise, patterns as best practices depend on the context, not on the individual. ▶Object 5 (p. 18) shows a proposed connection between terms of the behavior-setting theory and the pattern domain (refer to the glossary for definitions, p. I). Regarding forces, one should note that in the behavior-setting theory they are seen to generate or shape the synomorphs while in the pattern domain they need to be resolved through one of several creative solutions.

This kind of connection helps to understand the behavior-setting theory better: patterns are developed heuristically, and so are the standing patterns of behavior in behavior-settings. Therefore, a milieu does not determine the behavior by inherent prescription but by the best practice that is to be found in heuristic manner. The behavior-setting theory observes and explains behavior-settings but does not cover their development process. In analogy to the way that patterns develop it is assumed here that new milieus provoke diverse and experimental behavior that then settles down more and more to the best practices found in this heuristic process, forming a new behavior-setting.

2.2 Abstract organizational models for collaboration

Summary. The importance of the organizational paradigm is shown, and the term “underlying organizational paradigm” is introduced. Two complementary and well-known organizational paradigms are presented in some detail: the linear and the agile one. Especially the study of the agile paradigm covers quite different domains: agility in the manufacturing industries and in software development. It ends up with a domain-generic verbalization of agility.

Introduction. The study of sociological models of collaboration (see ▶chp. 2.1, pp. 14) suggests that there is no immanent organizational structure in collaboration: ethnography discovers a multitude of living conditions, pragmatic action deals with a multitude of designed task environments, distributed cognition finds a great many task-supporting artifacts, situated action studies action as evolving from different situations, and behavior-settings are said to be self contained *designable* environments that determine behavior, including collaborating behavior.

So collaboration is a space open for design: it allows a design model that orders and structures collaboration. This is the space for organizational design and organizational paradigms. Every collaborating system has an organizational paradigm — if chaos is defined to be a valid organizational paradigm, too. And each CSCW application is tied to a specific organizational paradigm. While the details of designed organizational structures are relevant within the organization's scope only, the underlying organizational paradigms are abstract enough to provide generality. Here, different organizational paradigms are presented but the focus is on the agile paradigm; this provides theoretical background for searching the proper organizational paradigms for community CSCW applications later on[10].

2.2.1 Linear organizational paradigm

The linear organizational paradigm in current CSCW systems. CSCW systems are an invention for the workplace ▶[Hindus :a01, pp. 199–201$_{-198}$], ▶[Crabtree :a01, p. 265$_{-264}$]. Naturally, by supporting current project management practices, they are built according to an underlying organizational paradigm. Widespread task-oriented and project-oriented groupware systems exhibit this connection to a high degree while CSCW tools for unrestricted communication such as e-mail are sparsely tied to this paradigm by their nature. When agility arose as a counter-movement, it helped to explicate the yet prevalent organizational paradigm[11]. The distinctions made by agility proponents are utilized here to infer ten important organizational principles of what could be called the "linear" organizational paradigm.

1. **Hierarchical decision process.** Decision rights and interactions are centrally controlled. This was typical for large enterprises up to the arousal of agile manufacturing; cf. ▶[Brynjolfsson et al :a01]. Related concepts in CSCW systems are e.g. access rights, workflows, assignment of owners and responsibilities for tasks.

[10]However, no comprehensive overview of organizational paradigms is given here. This is not necessary, as one of the two presented organizational paradigm will later turn out to be adequate for community groupware (cf. ▶chp. 4, pp. 53).

[11]See for an example ▶[Howell et al :a01]. They recognize that there has been no explicit theory of project management ▶[Howell et al :a01, p. 2] and then provide one within a direct confrontation of traditional and agile project management.

2. **Hierarchical interaction style.** Non-hierarchical interactions are prohibited unless initiated by a common superior. This principle is identified and criticized by the agile design principle "non-hierarchical interaction" ▶[Dove :a02, p. 9].

3. **Fixed vertical dependencies.** The production of anything from the smallest components to the full products was integrated into an unchanging organizational structure; while modern enterprise organization emphasizes focusing on one's core competence and generating value through transient coalitions with other enterprises according to current requirements; cf. ▶[Brynjolfsson et al :a01]. A related concept in CSCW systems is the plan-driven approach of a project management which tries to vertically integrate all activities into one project, forbidding unanticipated on-demand coalitions between participating units.

4. **Fixed responsibilities.** The relationship between persons and their responsibility and activity is defined by roles. These roles are fixed. Agility turns against this: it demands that relationships are transient and dynamic and that the scope of activity has to be flexible ▶[Dove :a04, p. $9.6_{-9.2}$].

5. **Controlling, not responding.** Management in the style of Frederick Taylor may be compared to constructing the enterprise as a machine, with its employees as its mechanical components that work according to instructions; see also ▶[Brynjolfsson et al :a01]. This kind of management is the attempt to control the whole environment through plans, contracts and workflows and thus to nip changes in the bud; see also ▶[Howell et al :a01, p. 2]. While there are differences between work processes in Taylor's style and workflows ▶[Schwickert et al :01, pp. $3-9_{+2}$], both resemble the idea of plan-directed individuals instead of individual-directed plans. Thus, workflow systems are corresponding concepts in the CSCW area.

6. **Aversively motivated results.** If failure results in penalty, people are aversively motivated to avoid these penalties. They invent a multitude of verification, examination, acknowledgment and failure prevention

mechanisms; cf. ▶[Beck :a02, p. 40]. Kent Beck, in proposing the agile alternative▶[12], identifies this principle:

> No single action takes the life out of a team or a person more than being told what to do, especially if the job is clearly impossible. […] Along the way, a person told what to do will find a thousand ways of expressing their frustration, most of them to the detriment of the team and many of them to the detriment of the person. The alternative is that responsibility be accepted, not given. ▶[Beck :a02, p. 41]

7. **Predefined processes.** There are explicitly written procedures of steps to follow, including decision procedures that protect people's role-based decision rights. Kent Beck, writing for the domain of software development, characterizes this principle and its origins thus:

> The difference is between playing to win and playing not to lose. Most software development I see is played not to lose. Lots of paper gets written. Lots of meetings are held. Everyone is trying to develop "by the book," not because it makes any particular sense, but because they want to be able to say at the end that it wasn't their fault, they were following the process. ▶[Beck :a02, p. 40]

8. **Documented interactions.** Requirements, plans, knowledge, coding standards etc. all make their way into written documents. This is meant to provide conformity of results and a predictable and stable project execution. Confer ▶[Boehm et al :a01, p. 22–23$_{-21}$]. CSCW tools resemble this principle manifold where they are used to provide a full digital representation of relevant work.

9. **Quantitative control.** Confer ▶[Boehm et al :a01, p. 23$_{-21}$]. Statistics are employed to visualize the overall current state of a project. Counterparts in CSCW systems are project scheduling applications.

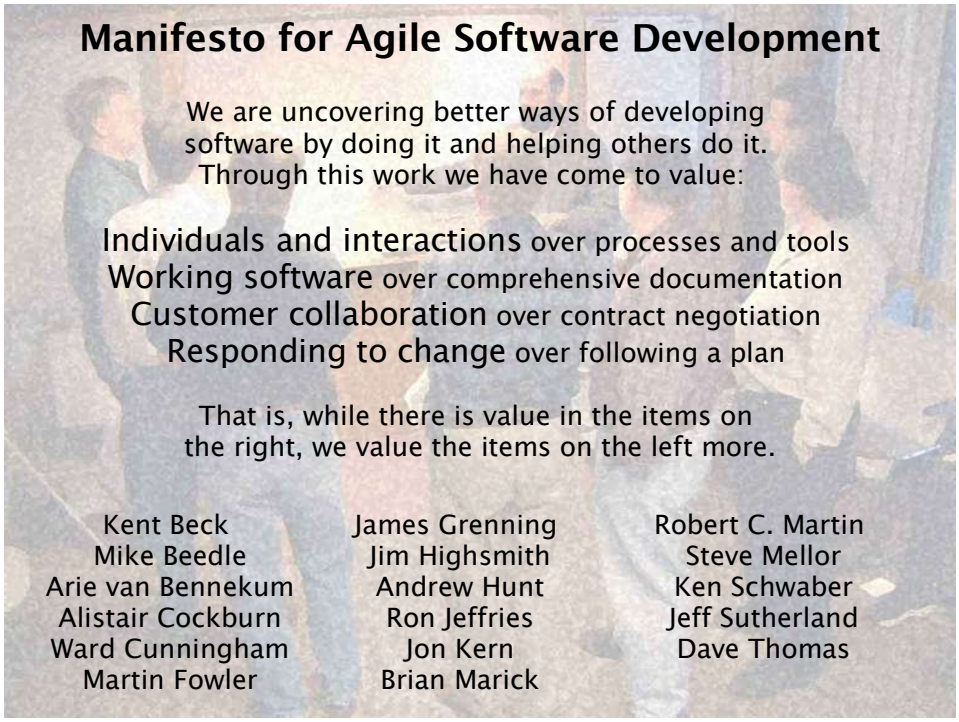
10. **Organizational invariance.** The home ground of the linear paradigm is a stable, only slowly changing environment ▶[Boehm et al :a01, p. 23$_{-21}$]. This corresponds to the fact that average CSCW systems offer few features to support the evolution of work practices, like supporting new conventions explicitly by tailoring the system. Work practices are assumed to be a part of the stable organizational environment.

[12] To allow oneself and other people to fail and to facilitate correction in good time with the team's help would get around this organizational overhead and culture of distrust.

2.2.2 Agile organizational paradigm

A short history of agility in manufacturing. The USA industry was cornered by Japanese lean manufacturing in the 1990's, which provided higher quality at lower costs. To cope with this competition, the USA started a Federal program. In its scope, the industry-led "Agility Forum"[13] studied change and change management from 1991 to 1998, producing knowledge about the agile enterprise and organizing, besides other things, the Annual Agility Conference. ▶[Dove :a01], ▶[Agility International :a01], ▶[Dove :a03, p. 1]

A short history of agility in software development. The object oriented paradigm and subsequent developments have inspired the development of agility in general from the early 1980s on ▶[Dove :a04, p. 9.14$_{-9.2}$]. However, a movement which applied agile design principles to the software development process itself appeared in public not before the late 1990s. Then, several agile methodologies had been developed, among them Extreme Programming (XP). A



Object 6: Manifesto for Agile Software Development, as shown in [Beck et al :a03]

[13]It was located at the Iacocca Institute, Pennsylvania, USA, which is affiliated with Lehigh University.

remarkable event was the 2001 gathering of a group that later named itself "The Agile Alliance". They agreed on a "Manifesto for Agile Software Development" (shown in ▶object 6, p. 23), finding the common core of the several agile methodologies ▶[Beck et al :a03].

A general definition of agility as used in the agile paradigm. History shows that agility was applied both in the manufacturing and in the software development industry. To grasp agility as an organizational paradigm one needs to find out what its multiple applications have in common. Rick Dove defines in context of manufacturing and general systems engineering:

> Fundamentally, agility is a reality issue. Things have always been changing, [...] but the pace and breadth of change have exceeded the response methods that once worked. [...] Agility fundamentally means confronting reality, in the business environment, in human behavior, and in technological infrastructure. ▶[Dove :a01, p. 1.3]

> Agile systems, as I define them, are concerned with response ability – for both reactive and proactive response needs and opportunities – when these are unpredictable, uncertain, and likely to change. ▶[Dove :a03, p. 2]

For the software industries, the Agile Manifesto (▶object 6, p. 23) seems to be the most encompassing definition. Another domain that is influenced by agility is project management; see for example ▶[Highsmith:a01] and ▶[Augustine :a01]. Abstracting from these domain-specific notions of agility, the following could serve as a general definition:

The agile organizational paradigm is about choosing the next step freely but sensible to pursue a moving objective in a changing environment. Due to the frequently changing conditions, classic approaches with detailed up-front planning, centralized decisions, rigid control, change-excluding contracts, determined methodology, technocratic conceit etc. are inappropriate or overhead. Agile organization abandons these, aligning all activity to the objective alone. It finds supportive means in domain-specific flexible techniques, often including teaming, quick feedback, rich communication, empowerment and reconfigurable modular systems.

Agile systems as complex systems. The behavior of agile systems is not determined by linear plans like in linear organization. Systems which allow non-linear interaction of their units can exhibit properties of complex systems ▶[Dugdale et al :a01, p. 1]. So agile systems can be considered to be complex

systems and it is little surprise that extreme collaboration (an example of agile collaboration) is found to exhibit the properties of complex systems ▶[Bellamine et al :a01, pp. 3–4].

The properties of sociotechnical complex systems are: non-determinism, limited functional decomposability, distributed nature of information and representation, and emergence and self-organization ▶[Dugdale et al :a01, p. 2]. The direct resemblance to the agile design principles "self organizing relationships" and "distributed control and information" (cf. ▶object 7, p. 26) is obvious. Additionally, complex systems have inherent change management mechanisms, again in parallel to agile systems:

> If a system is capable of self organisation, its functions evolve over time so that they can respond better to the requests of its environment. In this sense, a complex self-organised system cannot be described as structurally stable. ▶[Dugdale et al :a01, p. 7]

Connecting agile systems, CSCW and complexity theory seems a promising area of research but is sadly out of the scope of this thesis. For a start, cf. ▶[Dugdale et al :a01] and ▶[Pavard :a01].

Design principles for agile systems from the Agility Forum. Now, design principles of agility will be introduced and some characteristics of agility will be studied, abstracted from their original domain-specific context. The concept of agile manufacturing was defined by the Agility Forum. This organization was started in 1991 as a government-funded workshop at Lehigh University, led by Rick Dove. One of its results was the identification of ten design principles for agile systems as shown in ▶object 7 (p. 26). Rick Dove assures that these principles are "system generic", apt to bring adaptability resp. agility to every system ▶[Dove :a04, p. 9.3$_{-9.2}$]. He affirms the empirical origin of these principles while attributing the conceptualization to object-orientation[14]:

> The ten Rrs design principles [...] grew from object-oriented concepts, and have since been augmented with understandings from production and enterprise systems which exhibit high degrees of adaptability. ▶[Dove :a04, p. 9.15$_{-9.2}$]
>
> The ten principles employed here have been discovered, refined, and validated in numerous analytical exercises [...]. We have found useful

[14]This is observable in earlier publications of Rick Dove where the principles are named in more object-oriented manner: "Encapsulated Modules, Plug Compatibility, Peer/Peer Interfacing, Loose Coupling, Distributed Control/Information, Self Organization, Scalability, Redundancy, Reusability, Promiscuity" ▶[Dove :a02, p. 9].

repeatable patterns that appear to govern adaptability. ▶[Dove :a04, p. 9.14–9.2]

Self Contained Units
System composed of distinct, separable, self-sufficient units not intimately integrated.

Plug Compatibility
System units share common interaction and interface standards, and are easily inserted or removed.

Facilitated Re-Use
Unit inventory management, modification tools, and designated maintenance responsibilities.

Non-Hierarchical Interaction
Non-hierarchical direct negotiation, communication, and interaction among system units.

Deferred Commitment
Relationships are transient when possible; fixed binding is postponed until immediately necessary.

Distributed Control and Information
Units respond to objectives; decisions made at point of knowledge; data retained locally but accessible globally.

Self Organizing Relationships
Dynamic unit alliances and scheduling; open bidding; and other self adapting behaviors.

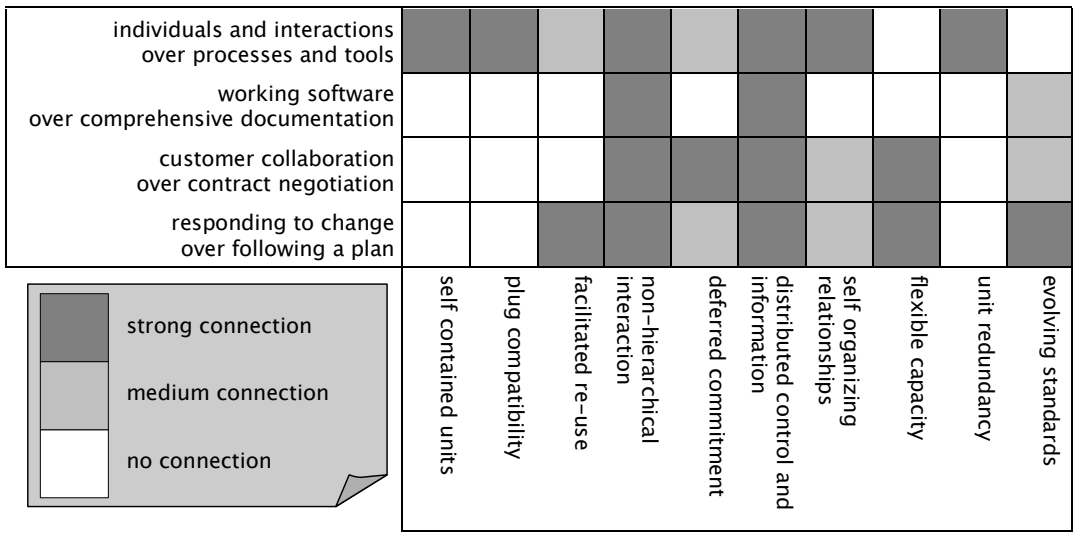
Flexible Capacity
Unrestricted unit populations that permit large increases and decreases in total unit population.

Unit Redundancy
Duplicate unit types or capabilities to provide capacity fluctuation options and fault tolerance.

Evolving Standards
Evolving open system framework capable of accommodating legacy, common, or completely new units.

Object 7: Agile design principles from the Agility Forum [Dove :a04]

Design principles for agile systems from software development. There is a multitude of agile software development methodologies just as there is a multitude of processes in the manufacturing industries. From manufacturing, a common set of ten agile design principles emerged (cf. ▶object 7, p. 26). To search for equivalents in the software domain is to search for the "common ground" of agile software development methodologies. The Agile Manifesto (▶object 6, p. 23) serves as such. It is quite obvious that its principles are nearly an alternative verbalization of the agile design principles as these both sets exhibit strong correlation (see ▶object 8, p. 27). The Agile Manifesto is however tied to the social domain, i.e. less system generic.



Object 8: correlation of Agile Manifesto principles with agile design principles

The XP principles define an agile system in the software domain. Kent Beck defines five basic and some less central principles underlying XP, his agile software development methodology. Though coherent with the agile design principles they are not system generic but they implement agility into a social system in the software development domain. Likewise, there are case studies about how agility was implemented in manufacturing companies ▶[Dove :a04, pp. 9.6–9.13$_{-9.2}$].

This means that implementing the agile design principles is creative and adaptive — being agile alone does not guarantee that the system does what it is intended to do. There is no universal agile system which can do anything whatsoever. An agile system has fixed parts (the framework) that need to be designed in correspondence to the environment; an agile system will not work if those fixed parts of the environment change.

Though Beck's XP principles are tied to software development, here are the more general of them from ▶[Beck :a02, pp. 37–42] in annotated form, to give a first impression of agile systems in practice. As this deals with an agile system in the social domain, it is quite relevant for developing agile groupware. Note that in XP the product is itself agile, as it must be highly adaptable; this case is unusual in the manufacturing domain and the CSCW domain. Now, the XP principles:

- **Rapid feedback.** Feedback makes it possible to adapt to changes, but the learning effect suffers greatly even from small latency between action and feedback.
- **Assume simplicity.** Provide simple solutions to your problems, not highly flexible, reusable, extensible solutions. Creating simple solutions is easier; changing them is harder but mostly unnecessary. Note that simplicity does not contradict the agile design principle "plug compatibility": if needed, the simple solution will be plug compatible. But plug compatibility is not necessary in all cases as not all environments are equally change-intensive.
- **Incremental change.** "Any problem is solved with a series of the smallest changes that make a difference." ▶[Beck :a02, p. 38].
- **Embracing change.** Solve your most urgent problem and simultaneously preserve the most options.
- **Teach learning.** Equip collaborators with expertise, not doctrines.
- **Play to win.** "The difference is between playing to win and playing not to lose. [...] Software development played to win does everything that helps the team to win and doesn't do anything that doesn't help to win." ▶[Beck :a02, p. 40]
- **Open, honest communication.** The environment must allow to discuss facts of reality as they are, without people getting affronted, enraged or in a huff.
- **Work with people's instincts, not against them.** "People like winning. People like learning. People like interacting with other people. People like being part of a team. People like being in control. People like being trusted. People like doing a good job." ▶[Beck :a02, p. 41]
- **Accepted responsibility.** Telling people what to do deprives them of their motivation to do it. The alternative is to let people accept tasks.
- **Travel light.** A changing environment forbids accommodating to its current disposition. Tools and artifacts might be convenient now and useless after changes. So stick to a few simple artifacts that are valuable beyond change.

2.3 Concrete organizational models for collaboration

Summary. Here is a small collection of implementation-related models and model-fractions to structure collaboration. Unlike the sociological models and organizational paradigms, they are concrete enough to support collaboration in CSCW systems directly; and hence are called "CSCW models" here also. They might be related to specific abstract organizational models (organizational paradigms). But even where this is the case, this resembles not their origin, which is connected with direct observations and not with abstractions mostly.

Awareness. One might classify current CSCW applications into those using the "tool paradigm" and those using the "environment paradigm". In the former, a CSCW application is a tool or a collection of tools, enabling individuals to start interaction with other individuals; e.g. e-mail falls in this category. However, such tools fail at providing awareness of the current situation and of the social context. This information is at hand for co-located collaboration but must be explicitly provided in distributed settings. Another problem related to the delivery of awareness information is: groupware usage may fade out if users need to log in to check for new information and then become dissatisfied by unsuccessful searches.

The importance of providing awareness initiated the development of the "environment paradigm" where a CSCW application is thought to provide an environment for collaborating activities, including a common situational and social context. See ▶[Pankoke-Babatz :a01, pp. 15-16$_{+16}$], also for more details. For examples of recent research work on concrete groupware awareness features see ▶[Gensel et al :a01], ▶[Fuchs et al :a01], ▶[Brignull et al :a01], ▶[Borges et al :a01]. A related concept are "coordinating representations", see ▶[Alterman et al :a08], and ▶[Alterman et al :a03].

Artifact-based models. Chengmao Xu introduced this innovative model in his doctoral thesis: ▶[Xu :a01, pp. 111-114$_{+12}$]. Xu introduces it as an alternative to the "tool paradigm" ▶[Xu :a01, p. 113$_{+12}$], i.e. something proving an environment for collaboration. The artifact-based model resembles the everyday working environment:

- **Multiple tools.** There are multiple tools available, each modeled by a coordination artifact and corresponding to a tool of traditional groupwares.
- **Information artifacts.** Just as there are documents and tools on a physical desktop, documents and tools are both artifacts.
- **Recursive containment.** Container artifacts allow the user to define containment relationships, e.g. structuring the groupware into rooms, workplaces and (private or shared) boxes.
- **Proximity definitions.** So-called layout artifacts are capable to define the relative proximity of artifacts within a container.

The artifact-based model allows fully flexible customization of the groupware, building "virtual offices". And just like an office, it allows individual and group work to occur simultaneously in a shared workplace (a synchronous group communication is automatically initiated by artifacts marked as "triggers", e.g. when people access the same room or document). Leaving the unfinished work laying around in the shared workplace allows a better asynchronous collaboration, as the artifacts give hints on the work's status. This latter notion connects the artifact-based model to the theory of distributed cognition, which deals with communicating through artifacts instead of by direct interaction only.

Another advantage of the artifact-based model seems to be the provision of a consistent metaphor, simplifying it to apprehend the system for users less skilled in information technology. The notion "artifact" comes from the real-world "thing" concept, just as the notion "object" in the object-oriented paradigm. This connection could make it feasible to use object-oriented methodology like design patterns for the configuration of artifact-based applications. And it even bears a link to agile system design because agile design principles are in turn derived from object-oriented system design ▶[Dove :a04, p. $9.15_{-9.2}$].

Workflow systems. A workflow formalizes routine interaction into a determined path of actions to be taken. Effectively, the CSCW tool controls the action. Workflow systems have been criticized for not respecting the improvisational and situated character of even planned action: plans *suggest* a way and leave it to the individual how to deal with unanticipated change. See ▶[Pankoke-Babatz :a01, pp. 13_{+16}] for a broader discussion. The desire to strictly

control action via static workflows is quite the opposite of the desire to adapt quickly to changes via agile systems. There is also research work on customizable workflows ▶[Frank :a01].

Conventions. Conventions and usage patterns are pragmatic means employed by users to fit CSCW applications better to their current needs. They are likely to develop in all recurring collaborative settings. Their big advantage is their flexible and self-regulated adaptability to change, without the need to change source code. Problems emerge where groups fail to develop sets of conventions or to commit to them ▶[Mark :a02]. Contextual awareness aids the process of convention development ▶[Fuchs et al :a01]. Conventions also play a role in the development of coordination patterns ▶[Alterman et al :a02], which bears a connection to the behavior-setting theory. As conventions are non-intuitive to new users, a registry for conventions might be a good idea to raise usability; or even, the possibility to support emerging conventions explicitly by configuring the software accordingly.

2.4 Model of community life

Summary. A detailed but general description of community life is provided and the importance of such a description is marked out. This model of community life was derived through extensive literature study and additional observations and covers organization, preferences, work and communication. The bottom line is that community differs from the workplace in many aspects which are important for the design of community groupware.

Introduction. A model of community life is necessary to inform the design of CSCW applications for communities. Note that home and community differ, but community life overlaps with domestic life when people interact with community members when at home; e.g. by talking in person or via telecommunication devices. This interdependence leads to accommodation on both sides, i.e. designing for community must take in some constraints of domestic life. Nonetheless, domestic life and community life still differ widely. For example, communities include more activity towards goals of shared interest. But homes prioritize resting, phatic communication and aesthetics ▶[Bayley et al :a01, p. 327$_{-324}$]. This thesis will propose a design for communities; this model however

includes also those facts of domestic life which are believed to be important for the community setting as well.

This model provides information needed to design coordination technology for communities, but may fail to present otherwise important or interesting features of community life. It is a focused model, no representative one. It will however prove quite useful as it is not that easy to find an equally detailed and inclusive model of community life elsewhere — this one was collected from many different sources. The model is presented now as a collection of fact statements about community, grouped around a topic.

2.4.1 Organization

- **Community has boundaries.** A boundary is necessary to define a community: it does not include all people, so there needs to be a criterion to decide somebody's belonging to the group. This criterion might be fuzzy, however. Or, as Alan Dix *et al.* puts it: "'Community' is based on a bounded and relatively small-scale set of relationships. However, the boundaries of community are not just spatial but also relational, social, technological, institutional etc.. This therefore incorporates some notion of 'membership', (and of awareness of membership) of inclusion and exclusion as well as ideas about apprenticeship, of 'learning the ropes' to become a member (or a 'stronger' member) of the community." ▶[Cheverst et al :a01, p. 2]
- **Communities are dynamic.** "Communities are dynamic and are always under development [...] and similarly the community is expected to endure. This might, for example, incorporate the provision of some sense of history through an archive as well as an orientation towards development and change." ▶[Cheverst et al :a01, p. 3]. The dynamics of community are also seen in the ever-evolving practices of community ▶[Adler et al :a01, pp. 215-217$_{-209}$].
- **Communities are persistent.** Occasion-based social groupings are not termed "community", i.e. community includes a sense of duration, transcending individuals' involvements. Resembling this, "Network communities are durable across time, users, and particular uses, providing an ambient and continuous context for activity. [...] This persistence

contrasts with communication channels that are mobilized for specific uses." ▶[Adler et al :a01, p. 211$_{-209}$]

- **Community is a collection of relationships.** There is a graph of meaningful and relatively persistent relationships in community. It serves as the infrastructure whereby norms for behaviors and values are created; first by mutual orientation, then by collective negotiation of the results. ▶[Cheverst et al :a01, p. 2]

- **Community is a power set of possible groups.** From own informal fieldwork I may contribute that varying subset groups of community members are an important characteristic. Such subsets might exist exclusively (like a plenum) or parallel to others; they might be persistent or volatile; they might form themselves by organizational intent or spontaneously in communal spaces; they differ in fluctuation frequency; they differ in size and various other properties. But they share a social character (consisting of more than one individual) and therefore a need for communication which might or might not profit from electronic communication and coordination tools. The design of such tools might in turn benefit from the concept of community subsets; for example, each task bears an connection to a community subset, namely to those members of a community who participate in the task.

- **Social groupings are inseparable from community.** "In this [virtualizing] view community is an achieved social construct of mutual ties, orientations and obligations. Thus, while the spatial and temporal character of community may differ and change, small scale social groupings of various kinds remain crucial to social life in various ways. These social groupings have always been produced in the face of shifting and interconnected social, geographical and technical relations and remain a crucial instantiation of community." ▶[Cheverst et al :a01, p. 1]

- **Community happens in the communal space.** Community often has a designated space that it occupies; examples include clubhouses, cafés, student pubs, university canteens, monasteries, a congregation's worship room, so on and so forth. ▶[Brignull :a05, p. 19–20$_{+3}$] offers a detailed discussion of terms and characteristics and connects it with Oldenburg's

"Third Place"[15]: the first place is at work, the second at home, the third a place for informal public life. And even distributed communities have their communal space, namely the virtual spaces provided by communityware. In any case, community cannot be fully integrated into the routine of work or domestic life — it occupies its own location in time and space. This is here regarded as inseparable from community.

- **The participatory decision process within communities.** While the degree varies, all community members are welcome to make their contribution to decisions in progress. This applies to families as well, but is different for hierarchical governmental and commercial organizations. For negotiating single calendar events this means that collaborative negotiation is an important point whereas predicting event attendance etc. is more relevant in the work context ▶[Crabtree et al :a11, pp. 132–133$_{-118}$].
- **"Families are not organizations.** In the past 50 years, the study of families has been the purview of sociology, and there is a large literature on family dynamics and home life […]. Family structures are complex and not hierarchical, at least not in the sense that corporate organizations are structured. Decision-making and value-setting are quite different within households." ▶[Hindus :a01, p. 201$_{-198}$]. These special circumstances must be taken into account when designing CSCW technology for the home.

2.4.2 Preferences

- **Work is not of highest priority.** While communities have shared interests, activity and performance is not its primary concern, as it is in commercial settings. Especially where community intersects with domestic life and the private area, it is about social relationships, fun, resting, beauty and self-realization. ▶[Bayley et al :a01, p. 327$_{-324}$]
- **Emphasis on simplicity of technology.** Simplicity in operation includes low system complexity, singleness of service, convenience, high service quality, simple user interactions, high usability and reliability. See ▶[Bayley et al :a01, pp. 327.329.331$_{-324}$] for its importance. It is my personal interpretation that to require simplicity of technology is the luxury of the non-commercial part of life: one might do so even if it is not

[15] introduced by Ray Oldenburg in ▶[Oldenburg :a01]

efficient. Simplicity as not requiring mental effort corresponds to the character of home as a place of rest. Simplicity raises effective simple means (like handwritten notes) over high-tech multimedia tools if the latter are hard to use, limited in service quality or ugly ▶[Bayley et al :a01, pp. 327.329$_{-324}$]. Mark Weiser advices to beware of information overload to make "using a computer as refreshing as taking a walk in the woods" ▶[Weiser :a01, p. 104$_{-93}$]. An advice from the Casablanca project is:

> Express just enough meaning, but not too much. Designers need to respect the value of perceived simplicity as well as the need for enough information and for expressiveness on the part of users. ▶[Bayley et al :a01, p. 331$_{-324}$]

- **Emphasis on aesthetics and fun in use.** See ▶[Bayley et al :a01, pp. 327.329.331$_{-324}$]. Demanding for these non-functional attributes is a luxury of the non-commercial part of life and corresponds to the character of home and of "third places" (like communities) as places of rest.
- **Community as a display of self.** ▶[Hughes et al :a01, p. 34$_{-24}$] observed in the fieldwork "the obvious but important point that households are more than simply utilitarian arrangements for living, but also the setting for a whole series of claims about 'identity' 'style' and so on." This point remains the same for community settings, only the spatial means to express it might be more restricted. ▶[Cheverst et al :a01] exemplify this for a climbing community, where the climbers' guidebooks served as a "display of self".
- **Privacy matters in private life.** People do not like surveillance or monitoring technology in their homes, even if the information gathered is just the presence status and it is just transmitted to friends ▶[Bayley et al :a01, p. 329$_{-324}$]. From the Casablanca Project: "Another promising [research] topic is that of homes as sanctuary; privacy concerns within households arose in a number of ways. We were surprised at the subtlety of those concerns, and see this as a challenging aspect of designing new communication technologies." ▶[Bayley et al :a01, p. 331$_{-324}$].
- **Emphasis on resting times.** ▶[Smith-Berndtsson et al:a01, pp. 12-13] concludes from ethnographic fieldwork that it is not obvious resp. unlikely that "ordinary" people would like to have mobile time-saving and efficiency services like e-mail, voice meetings and online shopping. They

argue that e.g. parents have other priorities: instead of doing what can be done later in time they like to relax from their straining tasks while driving alone in a car. However, Smith-Berndtsson *et al.* acknowledges the interdependence: new technology might change these patterns, and other patterns might evolve to cope with the new situation.

- **Keep new obligations away from home.** The home is considered as a resting place. Technology will be accepted where helpful, but if it introduces new (social or organizational) obligations this might be a reason to reject it. One such obligation could be social pressure towards more communicative acts, initiated by new communicative possibilities. Compare ▶[Bayley et al :a01, p. 329.331$_{-324}$].
- **Successful designs for community respect the legacy.** The wide range of today's technical possibilities should not lead to radical technocratic solutions which arrogantly ignore the "here and now" situation in home and community. Even if one invented a revolutionary technical system that is in itself enjoyable for humans and of great benefit to them, it would simply not be adopted if it ignores the legacy by not providing a smooth transition path. History shows that the home (and likewise community as its functional extension) changes gradually over time. See ▶[Crabtree et al :a02, p. 1]. Crabtree *et al.* insists that this is a key point to voluntary adoption: "Indeed, the success or failure of technological innovations for the home might be seen to rest on their fitting into and adding value to the current historically constituted needs of domestic life" ▶[Crabtree et al :a02, p. 1].
- **Technology should be inexpensive.** Confer ▶[Bayley et al :a01, p. 329$_{-324}$]. A reason for this preference seems to be that time efficiency and thus amortization of technology do not have such a high priority as in commercial organizations. Personal monetary means are often quite limited, so technology (in the form of "just a utility") should not be expensive. In this perspective, the ideal case for community groupware would be to re-use existing hardware like mobile phones and to be available as a cost-free web service.

2.4.3 Work

- **Low emphasis on efficiency.** Compared to commercial organization, households and non-profit organizations with community character put relatively low emphasis on time efficiency (at least they do not try to measure productivity as done in many manufacturing environments). This is possible because time is not considered to be a money equivalent here.
- **Homes are not workplaces.** Debby Hindus mentions and elaborates this point in ▶[Hindus :a01, p. 201–198]: dwelling houses are not constructed as workplaces in the way industrial buildings are. They are not built for easy integration of new technologies, seldom equipped with pre-installed data networks, must rely on do-it-yourself installation of new technology and they must not be harmful to babies, children, elders or pets. This restricts the way that existing CSCW technology can be incorporated into homes.
- **Understanding home as a workplace.** "Given ethnomethodology's analytic orientation it becomes possible to appreciate that the home might be understood as a 'work' place in the mundane sense of practical action that all household members engage in to accomplish everyday activities in the home, whatever those activities may be." ▶[Crabtree :a12, p. 4]. The same seems to hold true for community: there is a multitude of practical action which is necessary and done just to accomplish activities the community decided to perform.
- **The rhythm in community and home.** Both in the life of families (esp. on weekdays) and in community life there is a considerable amount of "routine" or "rhythms", e.g. weekly events and meetings. Planning is needed just for single and exceptional events. Compare ▶[Smith-Berndtsson et al:a01, p. 8], ▶[Cheverst et al :a01, p. 1]. This rhythm takes place in social interactions and other activities, and both are relevant for community-supporting technology: Alan Dix *et al.* remarks that "the highly predictable rhythm of everyday activity sets the grounds for shared expectations and comprehension of behaviour – successful communities carry intelligible rhythms of interaction and awareness – which vary according to the community […]." ▶[Cheverst et al :a01, p. 4]. From another perspective, daily routine is a set of actions, designed "on the fly", to meet daily demands ▶[Hughes et al :a01, p. 33–24].

- **"Consumers are not knowledge workers"** ▶[Hindus :a01, p. 201–198]. Here, Debby Hindus identifies another issue to heed when migrating CSCW to the home: "That is, motivations, concerns, resources and decisions can be very different from those found within workplaces. Buying behavior is perhaps the most compelling difference. Consumers make purchases based on aesthetics, fashion, and self-image in addition to practical considerations of cost and utility. In workplaces, buying decisions are driven by productivity concerns." ▶[Hindus :a01, p. 201–198].

- **Action within communities is mostly pragmatic.** Richard Alterman *et al.* wrote an interesting paper "Pragmatic Action" ▶[Alterman et al :a10] on the everyday task environment that discusses complementary features of task environments and individual psychology. The principles apply seamlessly to most community settings as well. See the glossary for the definition. Some personal informal fieldwork revealed that pragmatic action within community leads to some efficiency through adapting to the task environment. But a far greater degree of efficiency would be economically possible by optimizing the task environment, as it is often poorly designed or emerged without design. So pragmatic action does not lead to efficiency, it just adapts efficiently to circumstances whatever they are. Pragmatic action tries to arrive at a positive personal investment/result ratio, shying away from investments that amortize for community but not for oneself. A reason for avoiding a joint optimization of circumstances seems to be the lack of proper coordination tools.

- **Productivity loss in groups.** Munkes discusses the productivity loss of group work when compared to the sum of individual work ▶[Munkes :a01, pp. 13–15]. His discussion is founded on different psychological research work, especially ▶[Steiner :a01]. Quoting the latter, he categorizes productivity loss into the following groups:

  - **Motivation loss.** Effects that lower individual efforts of solving group tasks. Munkes mentions social laziness (arising from non-identifiability of the personal contribution) and the free rider problem (arising from rating the personal contribution to be dispensable to group performance) ▶[Munkes :a01, pp. 13].

- **Coordination loss.** Effects that come from sub-optimal integration of individual accomplishments. Munkes mentions the estimation expectancy (e.g.: anticipating group estimation lets people withhold ideas) and the mutual production deadlock (e.g.: in brainstorming, only one member can utter ideas at the same time) ▶[Munkes :a01, pp. 13].

- **Time is never empty.** "Labour saving devices such as vacuum cleaners have enabled people to achieve cleaner households but have not saved domestic labour time, as standards of cleanliness have increased proportionately." ▶[Dewsbury et al :a01, p. 2]. This could be extended towards a possible principle: the time for domestic life is per definition what remains from the day after work; so there is no point in trying to minimize this part of the day. Instead, people try to maximize the benefit of this time according to their personal wishes and goals.

2.4.4 Communication

- **Communication is not just coordination.** Differing from work situations, communication in community and home life has a large social aspect: maintaining social relationships. This is especially true for women. So operating coordination technology should afford possibilities for social communication to fit into community life.[16] [17] ▶[Bayley et al :a01, p. 327$_{-324}$]

- **Community interaction is often informal and for enjoyment.** The nature of communication in communal spaces is informal and often spontaneous, used for enjoyment, meeting people and "hanging out"; ▶[Brignull :a05, p. 19-20$_{+3}$] gives an overview of dedicated research. Tools for community support will need to respect and support this space for sociability where

[16] A good example for this is Casablanca's successful prototype CommuteBoard which is a whiteboard, shared between homes: "This ephemeralness, combined with the colored digital ink, engendered a playfulness and informality that users enjoyed. CommuteBoard was useful as well as fun [...]." ▶[Bayley et al :a01, p. 326$_{-324}$]

[17] From Mark Weiser's 1991 vision of ubiquitous computing: "By pushing computers into the background, embodied virtuality will make individuals more aware of the people on the other ends of their computer links. [...] Even today, people holed up in windowless offices before glowing computer screens may not see their fellows for the better part of each day. And in virtual reality, the outside world and all its inhabitant effectively ceases to exist. Ubiquitous computers, in contrast, reside in the human world and pose no barrier to personal interactions." ▶[Weiser :a01, p. 103-104$_{-93}$]

work performance does not count. For example, a community groupware should be designed to resemble the inviting and open nature of community's communal space.

- **Communities need multiple interaction styles.** It is an observation in network communities that users seek to have richer interaction styles than provided by the online framework; so face-to-face and telephone interaction are included, for example; cf. ▶[Cheverst et al :a01, p. 2], ▶[Adler et al :a01, p. 211–209]. Considering network communities as a subset of communities it seems that community members desire a richness of possible interaction styles, where, like in network communities, "[i]nteraction [...] is not tightly tied to a particular task or channel, but allows for different kinds of participation: peripheral, informal, formal, or serendipitous." ▶[Adler et al :a01, p. 211–209]. This fact can be explained by the desire to have an adequate representation of the social relationships on which a community is based on. Richness of interaction includes the demand for synchronous interaction ▶[Adler et al :a01, p. 211–209].

- **Emphasis on expressiveness.** While in commercial circumstances limited communication channels are accepted for performance reasons, in home and community the importance of maintaining social relationships makes users require natural and expressive means of communication. For example, talking in person is preferred to telephone calls. ▶[Bayley et al :a01, p. 327.331–324]

- **Multi-user aspect of community communication.** The communityware used in network communities allows multiple participants to establish communication settings with each other flexibly, and to define to what degree their communication shall be public ▶[Adler et al :a01, p. 211–209]. Network communities can be considered as a model of co-located communities: both are communities, i.e. they correspond in important aspects. So just like network communities, co-located communities provide a universal space for communication that can be divided and configured according to the members' wishes.▶[18]

[18]There are of course social constraints and expectations which render it impossible to divide the communication space with maximum freedom at all times. Thus, adding communication channels through technical means might be interesting, bringing the freedom of communication configuration to co-located communities as well. As in

- **Community is a technological space.** To maintaining a household means to utilize a great range of different technologies, employed for very different purposes ▶[Hughes et al :a01, pp. 33–34–24]. Likewise, all other areas of life including community life are massively influenced by technology in the western culture.

- **Community as a sociotechnical phenomenon.** On the social side, community is about creating, maintaining and employing robust social connections between people. This included at all times the utilization of technology for communication, coordination and information. One might think of the automobile, telephone, mass media, e-mail, memos for family members, web logging etc. ▶[Adler et al :a01, p. 210–209].

- **Social practices and technology are co-produced.** New community-supporting technology normally gets no exclusive role and does not change life fundamentally but rather is domesticated by the people to fit in the current structure of their lives. This takes place through modification, configuration and innovative forms of usage. This is, however, an interdependent (or: co-production) situation: at the same time, social practices change to adopt and integrate the new technology.[19] ▶[Cheverst et al :a01, p. 2]. Individuals in a changing technological environment are, so to speak, constantly searching better patterns of technology usage, including both new technology and new forms of usage. So every new technology contains an experiment which might result in unanticipated forms of usage; which are often of higher benefit than the originally intended usage. It seems that there are ways to design products which provoke such invented forms of usage; as an example, consider 3M's Post-it® notes. To study existing patterns of home life (like ▶[Hughes et al :a01]) should not limit the designer's horizon, creativity or courage.

- **The PC is yet undomesticated.** Prof. Alladi Venkatesh did ethnographic fieldwork on computer use in the home and observed that "many of the [information] technologies are incompletely integrated into the domestic life of the household. Specifically, the single-user desktop interface, and

network communities it is e.g. possible to participate in multiple public and private communication channels simultaneously.

[19]Reversing this produces a hint to identify where technological intervention into community life is helpful: identify social practices that are the attempted compensation of the *non-existence* of technical help.

the solo nature of computers in general, are inadequate for typical family use." ▶[Venkatesh :a01, p. 96$_{-86}$]

- **The idiosyncratic use of information markup in the home.** Crabtree *et al.* reports on the use of calendars in the home and identifies the user-defined annotations and notations of calendars as an important feature: it fulfills the current needs for markup and display fully and is nonetheless intelligible to the few other participants ▶[Crabtree et al :a11, p. 128$_{-118}$]. One will expect that such situation-specific, pragmatic, adaptively invented markup proves successful in small communities, too, where problems of understanding can be eliminated with low effort by communication. The benefits of the approach are the indefinite flexibility of such markup, including good change management capabilities; and that no formal teaching is required about how to use it.

- **The change-intensive nature of coordination.** Crabtree *et al.* observe through ethnographic fieldwork that calendars in domestic use are not fixed project plans but largely affected by eventualities and changing circumstances. This requires constant re-negotiation of affected events that are already scheduled, and the awareness of these changes to all participants who need to know them. Confer ▶[Crabtree et al :a11].

- **The distributed nature of communication in the home.** ▶[Crabtree :a12] and ▶[Crabtree et al :a07] report on the spatially distributed nature of communication devices, their usage in the home, and on "coordinate displays". (Coordinate displays are compositions of media to enable collaboration.) Though communities might have a different architecture of media usage, community relevant coordination acts are often performed from home. Therefore, new coordination technology for communities must integrate with the already existing coordination and collaboration devices in the home. Integration means higher efficiency than achieved by using both systems in parallel and to enter mutually relevant data redundantly and manually.

- **Minimize obtrusive communication acts.** Phone calls are considered obtrusive in home life. Replacing those needed for coordination only with unobtrusive communication devices like shared whiteboards will contribute to voluntary adoption ▶[Bayley et al :a01, p. 326$_{-324}$]. On the

other hand, ▶[Hughes et al :a01, p. 34$_{-24}$] reports that the telephone is a technology that seamlessly integrates with certain kinds of other activities. From own informal fieldwork I may contribute that even asynchronous communication is considered obtrusive where it interferes with synchronous communication; e.g. writing a lengthy SMS while on the phone or together with friends. This kind of intrusion can be nearly eliminated by making the asynchronous communicative acts very short, which requires appropriate input facilities.

2.5 Model of voluntary technology adoption

Summary. The importance of voluntary adoption and its character is pointed out, and "critical mass" is identified as one of its key factors.

Introduction. An artifact that is out of use is better than one never used: even trash is better than technology that got never adopted. CGW has the goal to obtain "delightful adoption", which goes beyond voluntary adoption in the sense that the new technology is esteemed as something desirable, valuable and enjoyable from the very beginning. But of course, voluntary adoption is the basis for this. The quick overview here draws from Harry Brignull's excellent and concise account in his doctoral thesis ▶[Brignull :a05].

CSCW adoption is a sociotechnical issue. ▶[Brignull :a05, p. 17$_{+3}$] presents technologies like the telephone and fax machine whose widespread adoption took up to decades, despite their usefulness. So technology adoption is an issue with non-technical factors; where multiple individuals are involved, there are social factors to be considered.

Mandated and voluntary adoption. ▶[Brignull :a05, p. 17$_{+3}$] distinguishes between mandated adoption and voluntary adoption. This distinction is somewhat blurred in community settings: while the community's decision to use a CSCW tool collectively is mandatory, it is the common and voluntary decision of its members; and though such a decision might be in place, individuals cannot be forced to comply as no monetary or otherwise strong leverage can be applied. In any case, it is the safe side to focus on voluntary adoption when designing for communities; just as done in ▶[Brignull :a05].

Only where people *prefer* a new technology, voluntary adoption takes place. Using a new technology exclusively within an evaluation period just because it was designed to exclude using the old technology in parallel is no indicator that people prefer the new one — rather, assume the contrary.

Critical mass is a key point to adoption in social settings. Obtaining critical mass is the best known problem with voluntary adoption of technology which is used by multiple persons ▶[Brignull :a05, p. 18_{+3}]. This thesis adheres to the view that the "critical mass" of CSCW tools deals more with the amount of useful information than with the amount of users. Another perspective to formalize the critical mass phenomenon is the "network effect", see ▶[Liebowitz et al :a01]. It looks at the gain in benefit contributed to a user's technology by the fact that others use the same technology (e.g. fax machines, e-mail etc.); this gain in benefit is called the synchronization value ▶[Liebowitz et al :a01, p. 671_{-670}].

The "network effect" perspective seems very appropriate for developing new communication technologies: to foster widespread distribution one should try to establish a medium value for the synchronization value. If it is too high, people will lack motivation to use the technology (at the beginning the synchronization value cannot be exploited as the user base is too small). If it is too low, people are not motivated to advertise the new technology to their friends.

From this basic idea of exploiting network effects, some design suggestions can be derived to foster quick and voluntary adoption of new communication technologies:

- **Respect legacy communication technologies.** At least the basic functionality of the new communication technology should be accessible from a broad range of communication devices. For example, it would do great harm to VoIP technology if it was impossible to call landline phones from VoIP phones and vice versa.
- **Remove synchronization value from the sending direction.** For example, e-mail-to-fax gateways could have been a good idea to foster e-mail usage in its early days. The same is desirable for the receiving direction but impossible in most cases.
- **Make it as easy as possible to participate.** Implement a new technology on devices that are already widely spread. Do not demand from customers

to buy new devices or to perform complicated installations of software. Make payment as convenient and as quick as possible. Develop apprehensible pricing models, especially flat rates. Use very moderate pricing for accessing the most important features of the new technology.

One should not overlook the social forces at work when dealing with adoption in social settings: technical benefit from network effects is not the only factor of adoption here. The social factors are covered by research on trend development and trend setting in areas like fashion and living.

3 Method: creative engineering, using abstraction and heuristics

> *Never regard study as a duty, but as the enviable opportunity to learn to know the liberating influence of beauty in the realm of the spirit for your own personal joy and to the profit of the community to which your later work belongs.*
>
> ***attributed to Albert Einstein***

Summary. This method seeks to find appropriate interaction patterns for CSCW in community use. It employs two foundational steps. First, an abstract organizational model apt for community use is determined (which includes a valuation of the model of current groupware). Second, this model is implemented into a concrete organizational model, i.e. into a pattern-based CSCW model apt for community use. The second step is creative and heuristic, consisting of iterated feedbacked inventions.

Justifying the target. As this thesis is located at the very start of engineering CGW, appropriate theoretic foundations are necessary. One target of this thesis is therefore a foundational model to build CGW upon. Such a model is not meant to explain something but to be *used* — it is a design model. There are design models on different abstraction levels (cf. ▶object 2, p. 11). Working towards an implementation in the end, a concrete one is inevitable: a CSCW model of collaboration.

Justifying the foundational steps. More abstract models are more general and are created by a larger research community. Therefore, to design a concrete model of collaboration it is good engineering practice to choose an abstract model first and then to implement it. With a CSCW model of collaboration as the target, the corresponding abstract model is an organizational paradigm (see ▶object 2, p. 11). From this result the foundational steps of the method employed in this thesis:

1. Determine an organizational paradigm apt for community use.
2. Implement this paradigm into a CSCW model for community use.

Step 1 in detail: towards an adequate organizational paradigm. As already stated, more generic models have a broader application and are developed by more people. Therefore, when searching an abstract organizational model for community use, it is likely to find one in the work of other researchers. The area to search includes all organization-related domains, e.g. organizational design, industrial engineering or software development methodologies.

Now, with such a multitude of candidates, how to decide if a given organizational paradigm is adequate? Just as with every organization, there is an organizational paradigm underlying community; however, it is not explicated yet. It is a legacy part of community as a complex social system; to change it, compensating changes are necessary or the system would collapse. But such radical changes are not possible here: design for voluntary adoption in community settings must respect the legacy (cf. ▶p. 36). Therefore, a community groupware must utilize the *current* organizational paradigm of community. Community life as it is measures if a candidate organizational paradigm is adequate.

The study of community life (cf. ▶chp. 2.4, pp. 31) provides enough ethnographic data to apply that measure. The ethnographic data was collected in foreign and personal ethnographic fieldwork and generalized up to the level that contains only the characteristics shared between communities. By applying these basic abstractions, the lengthy presentation of ethnographic data which would be necessary otherwise is replaced. Andy Crabtree's methodology of identifying patterns of home life (see ▶[Crabtree :a01] and the results in ▶[Crabtree :a08]) is somewhat parallel but more formal. Among the candidate models there will be the organizational paradigm prevalent in current groupwares; evaluating it will make apparent its shortcomings for community use.

This kind of design process is not unusual in the CSCW domain — it is called "ethnography in design" and elaborated upon on pp. 15. It is however unusual to transform the ethnographic data into an abstract model. In most cases, ethnographic data is presented as a detailed investigation into work and life practice, meant to provide a space for reflection, to evaluate one's concrete ideas for CSCW features. Of course abstraction in the sense of pattern recognition is done here, too — but just in the designer's head. Explicating the abstractions is often avoided, presumably because it is difficult. Here, this difficulty is avoided by searching and utilizing an appropriate ready-made abstract organizational model. The advantages of explicating this model are on one hand that it is

possible to communicate the results of the abstraction process to others and on the other hand that it provides clearer and simpler abstractions because an explicit model requires their creation.

Community's current organizational paradigm was developed in a long unconscious heuristic process by literally thousands of communities; it would be difficult for an alternative model to outperform it as a whole in community settings. The resulting community groupware will increase community performance through the new technical possibilities that it provides, not by exchanging the organizational paradigm.

Step 2 in detail: towards a CSCW model. Implementing an abstract organizational model for a concrete context requires creative systems engineering to fit the current context. There are case studies illustrating this for the agile enterprise (▶p. 27 resp. ▶[Dove :a04, pp. 9.6–9.13$_{-9.2}$]). These case studies show that the only general help that can be given in organizational design is to provide an appropriate abstract organizational model. All other help must is sensitive to the context, and that is necessary indeed: equipping governmental administration with car manufacturer's production cells would create agility, but for the wrong task.

To repeat what was said above: an abstract organizational model is appropriate for community if it is a framework to capture community's current way of organization without major changes. But: a concrete CSCW model is appropriate for community only if it positively supports at least most of the concrete elements of community organization. Because by doing so it supports efficiency.

In the first step, ethnography was employed to find an appropriate abstract organizational model. In this second step, ethnography is employed to find those more concrete and non-abstractable demands. Ethnography was adopted in CSCW design especially to make these demands visible to designers, as theories, models and taxonomies fail to do so:

> To provide a taxonomy […] — i.e., a scheme of classification — is not to make that work visible and available to design reasoning, it is only to classify a discrete ensemble of real world activities. What do those activities look like? […] Taxonomies don't answer questions like these because they do not show the work involved in the real world, real time accomplishment of discrete activities. Taxonomies make reference to real world activities — they talk about them — but they do not display the activities referenced in actual details of their accomplishment. […] It is, however, to recognise that as we do not know what calendar work

> consists of [...] then it is difficult for us to determine what appropriate design solutions might consist of concretely. ▶[Crabtree et al :a11, pp. 121–118]

Here, the study of community life (cf. ▶chp. 2.4, pp. 31) helps again. It has exactly the right level of generalization for this purpose: it shows what communities have in common. If it would be more abstract, concrete demands would be covered and the result would be an abstract organizational model as in step 1; this could lead to a general-purpose organization software but it would not support the more concrete common demands of communities and would be inefficient thus. If the study of community life would be less abstract, the application could not become a groupware for *all* communities.

Normally however, ethnography in design proposes to use ethnographic data without any abstraction, to make work practices visible to design as they are — see e.g. ▶[Crabtree et al :a11, pp. 121–118]. But work practices inevitably change with the introduction of new technology (cf. ▶p. 41) — we cannot spare our users from developing new work practices if we really want to provide new technology. Therefore it is useless and an impediment to innovation if one tries to preserve the actual work practices of current technology. These work practices will not even get studied here in detail, instead the following steps are employed:

1. Study the organizational paradigm and preserve it in the new technology.
2. Study what can be accomplished through the work practices which are used with the current technology and provide at least the same possibilities within the new technology.

These points are necessary for voluntary adoption, while preserving work practices is not: new work practices will develop quickly with intuitive and usable technologies.

The abstract organizational model of step 1 is also necessary. It is possible to invent single features without those principles[20], these features would not integrate to build up a software system however. So to end up with a system whose character as a whole is appropriate for community use, each of its features

[20]That is true especially because community is quite flexible; it can incorporates single tools even if they contradict its organizational paradigms. But it cannot adopt a whole system that contradicts its organizational paradigm: a CSCW application governs collaboration and would thus *change* community's organizational paradigm.

must be designed in accordance with both a concrete demand and the basic principles of community organization. Additionally, the organizational paradigm can serve as a quality measuring tool for invented features: derived from multiple existing systems, it is a formalization of what constitutes a good quality of software features for a specific domain.

This thesis' implementation of a concrete CSCW model of collaboration will consist of an interaction pattern collection in PLML format. These patterns are on the application level, independent of hardware platform etc.; however, they are concrete and complete enough to be implemented into a working community groupware without much creative efforts. Note that patterns might be used to model current practice, model current solutions, identify problems or propose solutions ▶[Crabtree :a01]. The latter is the usage of patterns proposed here: community life is not modeled in patterns but it is proposed in patterns how to support community life with CSCW tools. Though examined and optimized by means of ethnographic data, these patterns are proposals of solutions only: they have not been applied in community practice yet. Doing so will presumably lead to unanticipated appropriation of these patterns because it is usual for new community technologies and their patterns of actual usage to be co-produced (see ▶p. 41).

4 Results: the agile paradigm applied to CSCW

The locusts have no king, yet go they forth all of them by bands.[21]

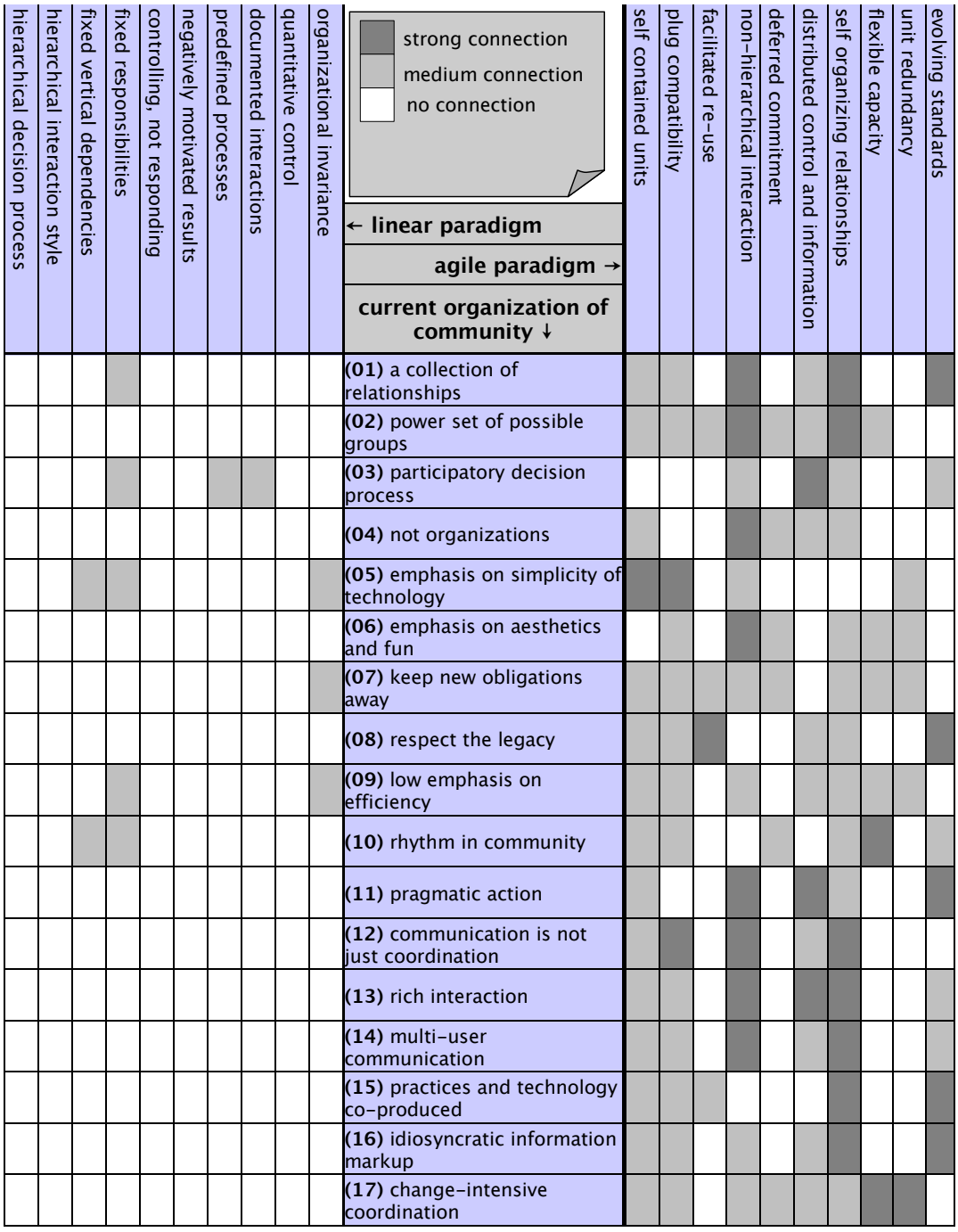
The Bible ▶***[ASV :a01, Proverbs 30:27]***

Summary. The following chapter describes the results from applying this study's methodology. Finding that the current organizational paradigm is inadequate for community use, it seems a good idea to repeat what was done in the manufacturing industry and in software engineering when thwarted by the rigid linear organizational paradigm: "embrace change" (as Kent Beck puts it in ▶[Beck :a02]), i.e. use the agile organizational paradigm. This is mapped afterwards to a concrete CSCW model in interaction patterns.

4.1 Agility: an abstract organizational model for community use

Summary. Shows the results from applying the first step of this study's methodology: agility as the adequate organizational paradigm for community groupware. The search for such a paradigm included the agile paradigm and the linear paradigm of current groupware applications. Both were evaluated for correlations with community's current organization; finding that the agile paradigm shows high correlation and is therefore an adequate formalization of current community organization and an adequate organizational paradigm for community use. The correlation is shown in ▶object 9 (p. 54), explained in subsequent paragraphs and then summarized.

[21]Rephrasing for systems engineering and organizational design: locust swarms are agile systems because locusts are *self contained units* using *self organizing relationships* to form swarms without any ruler, i.e. in *non-hierarchical interaction* ▶[Dove :a04, p. 9.6-9.2]. This viewpoint is quite funny and not even completely bogus: complexity theory contributes both to studies of animal micro societies and agile social systems ▶[Dugdale et al :a01, pp. 1-2].



Object 9: evaluating candidates for the organizational paradigm of CGW

On the correlation diagram. ▶Object 9 (p. 54) is meant to serve as a visualizing tool to find the underlying organizational paradigm of community organization. To get around the difficult and long process of providing proper

abstractions for community organization, two predefined sets of abstract principles are considered as candidates: the linear and the agile organizational paradigm, both of which are presented in ▶chp. 2.2 (pp. 19). Their correlation with current community organization is then evaluated. Correlation means either (for organizational aspects of community life) “is a possible implementation of”, or (for other aspects of community life) “is compatible with”. Only those findings from the study of community life (▶chp. 2.4, pp. 31) are considered that correlate at all with organizational principles — aspects orthogonal to organizational matters are irrelevant here. An organizational paradigm that exhibits a high correlation to community organization is then seen as a proper abstraction of community organization.

Of course, the decisions on the degree of correlation are hard to quantize and even qualitatively disputable in many cases; but no absolute measure is needed and subsequent explanations show that the decisions were made for all candidate models according to the same measure. Also, an absolute measure to estimate a candidate’s adequateness as organizational paradigm for community use is lacking. But it is not necessary either: ▶object 9 (p. 54) measures the candidates relatively to each other, and the one which scores highest will be used. While the number of abstract principles to which an aspect of community organization relates is no strong indicator (there could be “pure” implementations of only one principle), a weak or even no correlation at all is a strong indicator that this aspect of community organization contradicts an organizational paradigm.

On the degree of correlation. ▶Object 9 (p. 54) contains a multitude of decisions regarding the degree of correlation between aspects of community life and principles from two organizational paradigms. The more important and the less intelligible of them are explained in subsequent paragraphs, referring to aspects of community life by the numbers drawn in ▶object 9 (p. 54). This is connected with some insights into agility and community life.

The linear paradigm deprives of communication. The linear paradigm constrains interpersonal communication very much through its emphasis on plans, documents and a inflexible organizational structure which channels communication. Interpersonal communication is essential to community however (▶object 9 no. 12–14.16–17, p. 54). The need for rich communication in CSCW

tools for communities is also articulated in ▶[Adler et al :a01 p. 210$_{-209}$]; it facilitates to maintain a shared context.

The linear paradigm is too narrowly centered around work. Plans and their goals are central to the linear paradigm. But "[t]ask-focused or work-modeled connections can be too narrowly specialized to handle ad hoc and unanticipated group activities as well as evolution over time" ▶[Adler et al :a01 p. 210$_{-209}$]. While community includes planned action, other important community activities include personal relationship building (▶object 9 no. 2.12-14, p. 54), social groupings (no. 3) and fun activities (no. 6). The linear paradigm fails to support them while the agile paradigm allows the system to reconfigure itself accordingly.

The agile paradigm is adequate for pragmatic action. Agility and heavy-weight linear organization might be compared to craftsmanship and science: both are justified and coexist. Craftsmanship is pragmatic and low-overhead but cannot solve highly complex problems; the same is true for agility:

> But agile methods founder on handling complexity and to some extent conformity. They do not scale up to large complex projects, nor do they enforce obedience to order. ▶[Boehm et al :a01, p. 32$_{-31}$]

Science is highly structured and high-overhead and copes with complex problems. The same is true for the linear paradigm, where overhead is caused by documenting interactions, predefining processes and so on.

Community life is full of pragmatic action (▶object 9 no. 11, p. 54), which is best supported by agility:

- The projects to manage are situated in everyday life. They are far from highly complex projects, often consisting of single maintenance tasks or self-evident task sequences. For these tasks, planning is easy enough to be done in decentralized manner by individuals and just before execution. This is supported by agility as it focuses on individuals rather than processes ▶[Beck et al :a03]; formal methods with their high overhead are fully dispensable here.
- Agility can handle change-intensive environments (▶object 9 no. 17, p. 54). The everyday task environment is change intensive; thus, agility supports pragmatic action in handling this environment. In pragmatic

action, pattern development is the means to handle changing task environments (cf. ▶p. 16).

- Pragmatic action solves problems by drawing on the help of others where necessary ▶[Alterman et al :a10]; which resembles that helping each other out is an important element of community life. Agility supports pragmatic action here by providing free-form interpersonal communication (▶object 9 no. 12-14, p. 54). While the linear paradigm invests in lengthy documents that are meant as contracts but not for practical help.
- In pragmatic action, developing skill is accumulating experiences within the task environment, which includes to use the help that was coded into the artifacts of the task environment ▶[Alterman et al :a10, p. 54$_{-52}$]. This pragmatic skill acquisition is sufficient in the (relatively simple) task environment of community life. Agility is compatible with this as it abandons comprehensive documentation (cf. ▶object 6, p. 23).

Agility, just like craftsmanship, is in essence the absence of strict methodology in favor of a flexible space for interaction. As already shown, this is the organizational paradigm that supports community's pragmatic action. Pragmatic action itself is also void of strict methodology.

Community and agility share non-hierarchical structure. The linear paradigm exhibits hierarchy and centralization. In terms of traditional (i.e. linear) project management, hierarchical and centralized planning can be expressed thus:

> There is a managerial part and an effector part in the project; the primary function of the managerial part is planning, and the primary function of the effector part is to translate the resultant plan into action. ▶[Howell et al :a01, p. 3]

Community on the contrary exhibits a non-hierarchical structure and decision process (▶object 9 no. 03-04, p. 54); in typical scenarios there is simply nobody with the authority, qualification and time to develop central plans and dispatch their tasks. This makes community's organization incompatible with the linear paradigm but compatible with the agile paradigm which offers "non-hierarchical interaction", "distributed control and information" and "self organizing relationships". The importance of this aspect can be seen from the high

correlation of these principles with the whole spectrum of community life (cf. ▶object 9, p. 54).

Searching a role-free organizational paradigm. It is true that there are roles in community life just as in family life; they are implied in the differences of people's average occupation which is partially due to different preferences. The term "role" has analytic character here whereas it has a design character in commercial settings. In community settings, most tasks do not demand special profession and might in principle be taken over by any member, independent of roles[22]. This flexibility is not modeled adequately by the fixed roles and responsibilities of the linear paradigm but by the "self organizing relationships" of agility.

Community is an agile system of practice. The ten agile design principles (see ▶object 7, p. 26, and ▶object 9, p. 54) are system generic ▶[Dove :a04, p. 9.3$_{-9.2}$], i.e. they can be employed to build communities. These principles deal with systems of interacting units that share a common framework ▶[Dove :a04, p. 9.5$_{-9.2}$], which can be translated to systems of interacting people participating in a community. This system-theoretic perspective on people rather than just on technology may be unfamiliar but is promising: from this viewpoint, community organization exhibits a high correlation with the agile paradigm. This can be seen at a glance in ▶object 9 (p. 54) and indicates that communities are agile systems of practice:

- Community consists of independent individuals ("self contained units").
- An individual can collaborate with any other individual ("plug compatibility").
- Communication happens in an ever-changing network of interpersonal interactions ("non-hierarchical interaction", "self organizing relationships").

[22]Admittedly, this resembles also a personal ideal conception of community life, being highly efficient through full flexibility. In practice, explicit role assignments are found even in small communities of some types. This *could* be due to the lack of flexible tools for organizing people; as agile groupware is such a tool it is justifiable to adhere to role-free agility. This decision will change community life by added possibilities. Besides others, CGW will make all resources and information to perform role-free tasks available to all.

Community groupware could make community even more agile. Linear organization with its static hierarchy of sub-groups and fixed role-based responsibilities makes people manageable without CSCW support. It tolerates much inefficiency however: there might be idle and overstressed resources at the same time just because the static structure inhibits to help each other out. For the same reason, linear organization fails at spontaneous activities and handling external changes. One had to live with these consequences, at least what concerns coordination-intensive personal interactions, before information technology offered a revolutionary solution through electronic coordination support. Confer ▶[Brynjolfsson et al :a01, p. 246$_{-244}$].

We saw that community is already agile in character. So adding an agile groupware to it will hopefully boost its agility and efficiency by making more and fine-grained coordination possible, even in distributed communities.

Summary: correlations of community organization with organizational paradigms. It was shown that community organization can be described in terms of the agile design principles, i.e. community is an agile system. As these principles are *design* principles, they can be employed now to design an agile CSCW model in support of the agile character of community.

However, an agile CSCW model does not mean to just implement agile relationships between the application-level artifacts of community groupware. Community groupware is not an agile system of its own but shall merge with community into *one* agile system. Agility might apply differently to the information processing part of this system than to the rest. It will be rewarding to learn from other agile systems where coordination and information processing are important, e.g. to learn from XP how it deals with documentation, specification, communication and the like.

Though the agile paradigm correlates far better with community organization than the linear one, this is not to say that agility is the best paradigm for community organization ever. Just as in software development, the ideal solution will be a context-sensitive hybrid approach (cf. ▶[Boehm et al :a01]). Therefore, when designing a CSCW model from the agile paradigm, one should not hesitate to deviate from pure agility where it better fits community context.

4.2 MC³: an agile CSCW model for community use

Summary. Shows the Model of Computer-supported Collaborating Community (MC³ [23]) as the result of applying the second step of this study's methodology: inventing a CSCW model for community groupware. MC³ is shown in patterns and is meant to be a complete model to build a groupware upon, covering both abstract requirements (based on the agile paradigm) and concrete requirements (of community life).

Introduction to the MC³. Before presenting the MC³ in depth, here are the general characteristics of MC³-based community groupware. Such groupware will offer:

- expressive, informal, enjoyable and socializing communication, by means of conversation-style voice communication
- unobtrusive communication throughout the day, by means of unobtrusive voice messages
- ease of use, by means of low complex and intuitive communication features
- negotiation of the required synchrony and of the allowed obtrusiveness of communication
- mobile use from every current mobile phone; a speech-centric user interface affords convenient and fast input
- coordination which is seamlessly integrated into communication
- automating features for offloading quasi-mechanical steps of human coordination activity
- adequate design for the community setting

Architecture of the MC³. The MC³ is a model of collaborating community as an agile sociotechnical system — CGW is only a part of this system. This extended perspective on CSCW is comparable to the scope of the distributed cognition framework (cf. ▶p. 16) but in the domain of systems engineering. Using Rick

[23] Spoken ['em 'siː 'kyuːbd] and written "MCCC" also. And yes, there is an intended resemblance to the famous equivalent of energy.

Dove's key definitions for agile systems ▶[Dove :a04, p. 9.5$_{-9.2}$], the MC³ system architecture can be stated as follows:

- **The units are individuals.** Units are self contained, identifiable, interacting parts of a system; cf. ▶[Dove :a04, p. 9.5$_{-9.2}$]. In the MC³, all individuals of a community are units, and there are no other types of units.
- **The system is the community.** A system is a "group of interacting modules, sharing a common framework and serving a common purpose" ▶[Dove :a04, p. 9.5$_{-9.2}$]. In the MC³, the community is the system; as, a community is a group of interacting individuals.
- **The framework is CGW.** A framework is a "set of standards constraining and enabling the interactions of compatible system modules" ▶[Dove :a04, p. 9.5$_{-9.2}$]. In the MC³, the community groupware and community conventions define the interactions of individuals. The features of community groupware are shown in ▶object 10 (p. 65).

Viewing CGW's features as interactions between system units seems to disagree with the objective, unit-like character of its central feature (the pattern ▶"Agile message"). The apparent contradiction is resolved by this consideration: agile messages are *reified* interactions, i.e. interactions which are at the same time units of another system. This other system is stored in CGW and consists of agile messages interacting with each other (see pattern ▶"Interacting artifact"). One could argue that this other system must be agile, too, because it models the collaborating system and its changes over time. This reasoning is deficient, however: interactions facilitate unit communication, they are no model but a part of the agile system they occur in. Communication in the MC³ is ephemeral by default; and even where features help memorizing communication they should not be mistaken for models of (parts of) the agile system. MC³-based agile groupware does not contain features to build models of a system: no project records or plans, and no explicit support for tasks and appointments as concepts.

Viewing community as a system does not imply any static organizational structure. Community is modeled as an agile system and thus might change the configuration of its units totally while keeping its identity. Agile systems are not structurally stable, and agile enterprises are a good example for this quality ▶[Dove :a04, pp. 9.15-9.18$_{-9.2}$].

Character, format and types of the patterns. While the patterns propose all the central features needed the implementation of CGW, they were formalized as general as possible. Some had to be tied closely to CGW, but most are abstract enough to be useful in the whole CSCW domain and beyond.

This thesis employs PLML for the pattern format (cf. ▶p. 13). To reflect the "as general as possible" character of the MC³ patterns, the PLML problem section will contain what can be solved by agile design principles and the context section will contain the concrete requirements of community life. The context section should "characterise situations in which this pattern can be most usefully ('naturally') applied." ▶[Fincher :a02, p. 26₋₂₅]. So it does here: while the pattern can be applied wherever its problem arises, it was designed with the community context in mind and so fits best there. Concerning the patterns' names, they are chosen to express the solution, not the problem; this is the usual way of naming patterns, cf. the classical example of ▶[Gamma et al :a01].

To solve a complex problem, different patterns are combined. It affords a compact pattern language if building complex solutions is left to the user. However, the MC³ patterns are intended to be concrete enough for direct implementation into CSCW features. Therefore, some pattern combinations get explicated into their own patterns, solving CGW's complex problems. The mental work of selecting and combining these patterns becomes unnecessary this way. However, where combining the patterns is just a quasi-mechanical task, the result does not deserve a unique name and is indicated by a relationship only. While "combining patterns" add avoidable patterns to a pattern language, this does not imply textual redundancy as patterns are built upon each other.

There are more abstract patterns which facilitate comprehensibility through structure and which generalize common problems, reducing redundancy. Other patterns combine these abstract ones to solve composed problems. Only combining and stand-alone patterns are the proposed features for community groupware while the rest is re-usable in other contexts.

Some feature proposals are to be implemented not in software but in community's organization. These patterns for organizational design have been included because a more agile community will perform better with agile groupware. As work practice and technology are co-produced it could be assumed that these organizational design patterns would develop in practice

anyway, at least to some degree. But if communities know about these to-be-developed patterns beforehand they might replace experimenting with targeted organizational decisions.

How to justify the patterns. The agile paradigm was found adequate for community use in this study's first step and provides abstract requirements for the features of CGW. To justify a proposed pattern it is therefore sufficient to trace how it conforms to agile design principles. And additionally, of course, to reason how it fits the concrete requirements of community life. These concrete requirements are provided by the model of community life (▶chp. 2.4, pp. 31). The patterns' "evidence: rationale" part includes hints on the development process, so there is no need to recount it in more detail.

General characteristics of the MC³ pattern overview diagram[24].

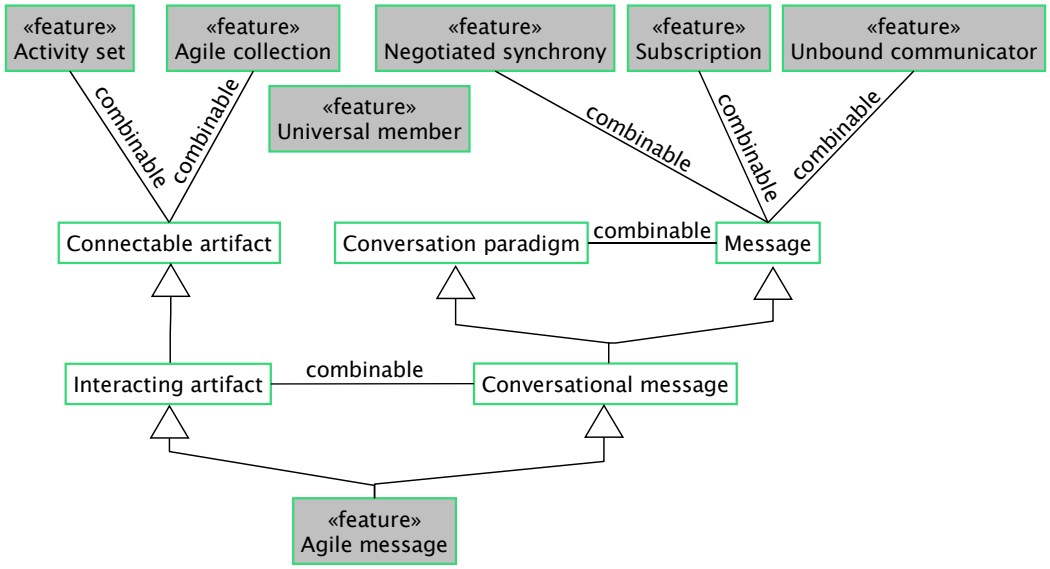
- **Diagram style.** Patterns are shown as classes in a conceptual UML class diagram; this is unusual as in object-oriented software engineering patterns are presented as sets of classes which have different roles. But it is justified: patterns are concepts *themselves*, however semantically on a more abstract level.
- **Representation of abstraction level.** From top to bottom, abstraction decreases continually in the diagram[25].
- **Feature stereotype.** Patterns which are the concrete feature proposals for CGW are distinguished by the stereotype "«feature»" and their background color.

Pattern relationships in the MC³ pattern overview diagram. The predefined PLML pattern relationships (cf. ▶[Fincher :a02, p. 27₋₂₅]) have been found to be somewhat shortcoming for the MC³ patterns. Therefore, the following relationships are used instead:

- **Combinable.** Indicates promising pattern combinations, in the sense of synergistic combinations. Some of these combinations have been explicated in the MC³ by patterns that inherit from two or more combinable patterns.

[24] See ▶object 10, p. 65.
[25] It is suggested to study the patterns in this order and according to their dependencies, not in the alphabetical order they have in this document.

- **Generalization.** The PLML "is-a" relationship "[m]eans that this pattern is the same as, or is an alternative solution to, the same problem" ▶[Fincher :a02, p. 27$_{-25}$]. This might be useful to relate pattern languages to each other. But "is-a" cannot express that a pattern re-uses and perhaps extends the ideas of another pattern, esp. that a more concrete pattern extends the ideas of a more abstract one. To express this, the UML generalization relationship is used instead. For combining patterns, multiple generalization is necessary; the generalization relationship is used for this in concordance with its meanings "extends" and "is a".
- **Aggregation and composition.** The PLML "is-contained-by" relationship "[m]eans that this pattern is 'smaller' and is used (with others) to instantiate a larger one" ▶[Fincher :a02, p. 27$_{-25}$]. Instantiation implies that the 'larger' pattern is abstract. In practice however, instantiating an abstract pattern leads to single patterns that extend it with more concrete functionality, in analogy to extending abstract classes in object-oriented software development. This relationship is already expressed by UML's generalization relationship. But one cannot express that a pattern "contains" another (in the sense of employing it to fulfill its own purpose, not in the sense of extending an abstract pattern). For this, the UML aggregation and composition relationships are used, depending on how independently the contained pattern might be used elsewhere.



Object 10: MC³ pattern overview diagram

4.2.1 Activity set

An activity set is a set of items which a user has opened simultaneously, and it is a subset of a navigable set. The items in an activity set are supposed to be mutually relevant, i.e. activity sets can be used in social navigation to recommend relevant navigational alternatives.

Problem. There is a large number of items forming an ever-changing graph of relevancy relationships. There is a number of people who explore these items. Each of them latently knows at least a part of these relevancy relationships. But they lack the time to create and maintain an explication of this knowledge, so they have to navigate between items by choosing the next item from a long lists of which most are irrelevant in the current context.

Context. Software applications which have their data used by multiple individuals. Examples include CSCW applications, where the items would be tasks, address book entries and the like. The pattern applies especially to an agile style of activity, which tries to avoid full documentation of knowledge because a large part of documentation might never be used.

Solution. Use social navigation: treat the steps that people take when navigating spatially as unconscious explication of their knowledge about relevancy relationships. Based on the acquired knowledge, recommend relevant items to users.

One factor that influences the recommendation is the current position of the user and its context. It has been suggested to use the path which a user went to arrive at an item as the context of the user's position ▶[Brodbeck et al :a01, p. 361$_{-358}$]. In paths, each item is directly connected with its predecessor and successor only. So it is impossible to derive one-to-many relevancy relationships between items from a single path. An item more than one step away in the path could or could not be relevant; looking at a linear path gives no hint to decide this. Therefore, allow the user to view multiple items simultaneously; in analogy to tabbed web browsing, users will tend to view mutually relevant items simultaneously. This transforms the path to a tree structure.

To detect unsuccessful navigational steps and thus to avoid false recommendations, restrict the number of items that can be viewed simultaneously. Before accessing another item, users will have to "make room", i.e. they will have to remove the least relevant item from the current set of items.

Additionally, the dynamics of navigation can be used to acquire knowledge about relevancy relationships. Dynamic aspects of navigation include: changing the active item of the set; adding items to the set; removing items from the set; the duration items remain in the set; and the duration an item is the active item of a set.

Evidence: Rationale. Social navigation is not a new concept; according to ▶[Höök :a01, p. 18$_{-16}$] it was introduced by Dourish and Chalmers in their short paper ▶[Chalmers et al :a02] from 1994. Since then, social navigation has developed a remarkable history of research work and applications, of which ▶[Höök :a01, pp. 17–18$_{-16}$] gives an overview. Chalmers *et al.* later introduced the path model ▶[Brodbeck et al :a01], a concrete means of social web navigation. This pattern extends these activity paths to activity sets by allowing the user to be located at multiple items simultaneously. But the central thoughts remain the same:

> By putting activity at the centre of representation and not the periphery, the path model concentrates on the reader not the author and the browser not the site. ▶[Brodbeck et al :a01, p. 359–358]

Limiting the number of items viewed simultaneously is an artificial, avoidable inconvenience; this decision deviates from designing user interfaces for usability only. It utilizes findings from the behavior-setting theory (cf. ▶p. 18): the behavior of individuals is significantly controlled by their milieu. Here, the design of the software environment causes users to close the least relevant items of a set.

This pattern is adequate to support interaction of individuals in agile collaboration. The Agile Manifesto values individuals and interactions over processes, tools and comprehensive documentation ▶[Beck et al :a03]; and so does this pattern ▶"Activity set". It does not result in explicit and complete documentation of relevancy relationships but draws necessary information from people's previous behavior. It is admittedly a software tool (something not valued high in the Agile Manifesto), but just to mediate interactions: navigating is equivalent to recommending relevant items without additional effort. Additionally, ▶"Activity set" implements agile design principles which deal with interaction (cf. ▶object 7, p. 26):

- **Non-hierarchical interaction.** ▶"Activity set" mediates interactions that deal with navigational help. It does this without a hierarchy, i.e. directly between individuals.
- **Deferred commitment.** ▶"Activity set" expects no permanent commitments from individuals, in contrast to manual maintenance of links.
- **Distributed control and information.** Social navigation collects information from many sources and thus acknowledges that navigational information is distributed among individuals. It does however not enforce individuals to record their navigational knowledge in a central database.
- **Self organizing relationships.** This means "[d]ynamic unit alliances and scheduling; open bidding; and other self adapting behaviors" ▶[Dove :a04, p. 9.6–9.2]. ▶"Activity set" is a self organizing mechanism for interactions that deal with navigation.

Related patterns.

- ▶**Connectable artifact.** ▶"Activity set" is combinable with all patterns that provide sets of items, as this constitutes material to navigate.

4.2.2 Agile collection

An intuitive, apprehensible, low-maintenance means to manage a change-intensive item set and to navigate it semantically. The set of items and its subsets are flat lists; subsets can include other subsets, which enables automatic change propagation.

Problem. When handling large numbers of change-intensive items in software applications, additional structure becomes necessary to maintain comprehensibility. It is however not obvious what might be an intuitive and usable solution here.

Context. Software applications that are used by people without formal training and with possibly below-average technical skills. The collections can be used for arbitrary purposes, e.g. in CSCW applications to memorize personally relevant tasks and messages in personal boxes.

Solution. Manage all items as one item set. Users may collect items into subsets. Collecting means adding items or whole subsets to subsets. Subsets and items have unique names. Present the item set as one flat list and allow users to filter the list by item or subset name. Subsets are dynamic: they immediately reflect changes within contained subsets, like the addition or deletion of items.

Evidence: Rationale. The solution is intuitive because flat lists resemble how non-technical people do their shopping lists. It is easier to navigate than tree-style subdivisions of tasks because the latter introduce depth as a second navigational dimension, raising system complexity.

▶"Agile collection" is adequate to support interaction of individuals in agile collaboration (cf. ▶object 7, p. 26): as a means to organize non-hierarchical communication it facilitates complex interactions which last for some time. In agreement with the agile design principle "deferred commitment", agile collections impose no fixed bindings between individuals but have good change management capabilities. Thus, they are in no way restricted to

support only linear up-front task decomposition which is typical for traditional project management ▶[Howell et al :a01, p. 2-3].

Tree-style subdividing works top-down while collecting items works bottom-up. The latter resembles the natural order of steps: finding additional structure to order already existing items. In the everyday task environment for example, users have no problem to recognize their atomic tasks but to remember them; decomposing aggregate tasks into atomic tasks is so obvious here that the decomposition is virtually co-existing with its aggregate task, assigning to the latter the role of just a collection of atomic tasks.

Related patterns.

- ▶**Connectable artifact.** ▶"Agile collection" is combinable with all patterns which result in item sets, e.g. with ▶"Connectable artifact" and all its derivatives.

4.2.3 Agile message

The central feature for community groupware, perhaps re-usable for other applications. It integrates communication and coordination seamlessly: coordination is effected through communication and supported by some automated communicative acts. This pattern respects the requirements of community life to a high degree. It results in an application that runs on mobile phones and relies on voice input for the most part.

Alias. Voice paper

Problem. As CSCW tries to automate coordination it needs a detailed representation of the situation to be coordinated. Making this information machine-understandable is costly in terms of labor and time. This becomes a major problem for distributed collaboration: mobile devices offer only limited input capabilities. Even worse, agile collaboration increases the volume of items to be coordinated because the item granularity is more fine-grained.

Context. Fully automatic coordination is simply not adequate for voluntary adoption by communities even if it pays off in terms of time. The reasons for this are that community attaches comparatively low value to efficiency (cf. ▶p. 37), instead prefers simplicity of technology (cf. ▶p. 34),

expressive, informal communication (cf. ▶p. 40), enjoyable communication (cf. ▶p. 39) and interpersonal, socializing communication (cf. ▶p. 39). This pattern proposes an alternative to fully automated coordination that fits community context.

Solution. Coordination should remain an essentially human task. This makes it unnecessary to machine-understandably describe what to coordinate. Humans should be supported in their coordination task by facilitating distributed communication (see the pattern ▶"Conversational message") and by offloading stereotypical mental activities to the CSCW tool (see the pattern ▶"Interacting artifact"). This pattern is basically a seamless integration of these both patterns, along the following outline:

- **The two roles of an agile message.** An agile message can be both a conversational message and an interacting artifact. This is implemented by attaching the interaction ▶"contribute artifact to group conversation" to an interacting artifact, which makes it a conversational message.
- **Linking communication and coordination.** Agile messages can be utilized as both communicative messages (comments, questions, …) and coordination items (tasks, appointments, …). There are several artifact connections to connect agile messages, differing in their semantics: citing them, replying to them, associating them, receiving reminders from them or creating several of them in succession. These connections are bidirectional; they can be used to connect communicative messages and coordination items during conversation. Then afterwards, relevant communicative messages can be accessed from coordination items. And from each communicative message one can navigate the past conversation in temporal or threaded[26] order, via connections to the preceding and the following communicative message.
- **Mnemonic text part.** Agile messages have a text and an audio part; one of these might be missing. The text part contains a limited-

[26]A "thread" means here a sequence of messages on a subject, not a tree as in newsgroups. A thread is continued by replying to its last message (and only this message offers the possibility to reply). A thread is created when creating a message that is not in reply of another. The pattern ▶"Conversation paradigm" enforces "serialized" conversation, so the several threads are interwoven.

length mnemonic title to represent artifacts when shown within visual coordination aids or as textual reminders. It might also contain information which is better to share as text than as audio, like URLs, phone numbers and names.

- **Audio part.** Agile messages have an audio part; it contains all content except a mnemonic title.

Evidence: Rationale. This rationale goes along the individual paragraphs of the solution, then adds some more general considerations.

Coordination as a human task. The effort to enter machine-understandable information can make a feature uneconomic and thus useless. This is true for fully automated coordination due to limited efficiency of current input technology, esp. mobile technology. Additionally, coordination automation like automatic scheduling is in its infancy. For these reasons, coordination should remain a human task. CSCW tools in support of *human* coordination will not *do* anything on their own, instead they will facilitate communication, awareness and memorizing. In this domain, they can safely ignore the semantics of coordination, getting around the overhead of formalized machine-understandable input.

Lowering the semantic understanding of CSCW tools is against the trend but reasonable until revolutionary input technology is developed. An example: where CSCW tools schedule appointments automatically, people will need to enter their free and occupied times and perhaps preferences and other information. This amounts to a lot of input activity. The groupware will then make up appointments, which might require complicated solutions like preference-based group scheduling ▶[Brzozowski et al :a01]. On the other side, the agile way of making appointments is to provide a reliable and unobtrusive medium for negotiating appointments in social discourse. Instant messaging affords this ▶[Nardi et al :a01, p. 81$_{-78}$] but is not as reliable as a mobile application will be.

Where coordination remains a human task, CSCW tools might offer automation anyway, namely for social communication, awareness and memorizing▶[27]. These features (for automating aspects of conversation) can

[27]These features are provided by the interactions for agile groupware in pattern ▶"Interacting artifact".

take over some quasi-mechanical activity, both mental and communicative. But they are unable to make coordinative decisions or to intelligently respond to objectives, for the reasons stated above. However, they can be utilized to compensate for some drawbacks of human conversation: its ephemeralness, invisibility to third parties, low efficiency for recurring tasks, and its lack of visual coordination aids (cf. ▶[Brignull et al :a01] and ▶[Alterman et al :a03] for these points).

The two roles of an agile message. Coordinating representations like calendars, pinboards and to-do lists contain small informal messages (also cf. p. 42). The same is true for conversation: it contains messages of limited length and without formal structure. So the "data type" is the same in both cases. In artifact-based groupware, one artifact type can be used for both functions; in this case, agile message. This feature consolidation makes it possible to re-use a coordination item (like a task) as a contribution to conversation, e.g. to make the group aware of a newly created task. And vice-versa, it makes it possible to re-use contributions to conversation as coordination items; compared to face-to-face communication, this removes the necessity to restate ephemeral conversational statements in non-ephemeral ways, like to write down a task.

The possibility to treat coordination items and communicative messages alike is limited by the ▶"Conversation paradigm" pattern, however. It demands that a conversational message is only accessible by those who attended the conversation when the message was first uttered. Coordination items like tasks and appointments however will be visible to the whole group. This is achieved by adding interactions which include the artifact into a coordinating representation (cf. ▶p. 90).

Linking communication and coordination. Applying the ▶"Agile message" pattern causes coordination to be accomplished through communication. This shifts most information to conversation, rendering it sufficient if coordination items like tasks consist of a mnemonic title only. To save the information conveyed in conversation which is related to a coordination item, conversation is automatically stored and linked[28] with the concerned item. Links are created whenever a coordination item is accessed in

[28] implemented as artifact connections (see pattern ▶"Connectable artifact")

conversation. This affords a seamless integration of coordination into communication, emphasizing the communicative and interpretive use of calendars and probably other coordinating representations, at least in domestic life ▶[Crabtree et al :a11, pp. 124–129$_{-118}$].

Mnemonic text part. In home and community life, to-do lists and calendar entries consist of some mnemonic keywords only. They are no full representations of tasks and appointments but just utilities to take over mental efforts like memorizing and coordinating. So it is entirely sufficient if agile messages consists of a mnemonic text part only, without an audio part. This requires however that all recipients are made aware of the message's meaning through a common context, within group conversation or in other ways.

Audio part. Spoken natural language is a natural, expressive, convenient and fast input method with very low formal overhead. It is appropriate for communication and also for coordination: the fastest way to create a task is to utter it aloud. It is probably the most convenient input method possible on current mobile phones (using mobile phones is favorable to mobile community groupware, cf. pattern ▶"Unbound communicator"). Agile messages contain both text and audio, so handling them is best supported by a multimodal interface. The desired sequential multimodal input is supported by WAP-enabled mobile phones ▶[Lo et al :a01]. More integrated forms of multimodal input are under development for 3G mobile phones ▶[Finan et al :a01, p. 109$_{-102}$].

Justifying ▶"Agile message" from agile design principles. ▶"Agile message" is MC³'s central pattern. It is adequate for agile groupware if it affords an interaction style for agile collaboration of individuals, just like face-to-face interaction does. From the perspective of systems theory, this can be stated as: if it provides an interaction type for agile interaction of system units (cf. ▶p. 60)[29]. Some agile design principles (▶object 7, p. 26) serve criteria for agile interaction. The correlation with these principles is examined thoroughly now:

[29]At the same time, agile messages *are* system units (forming a graph of connected artifacts). Viewing them as reified interactions resolves the apparent contradiction (cf. p. 60). Agile messages support agile collaboration if they exhibit agile characteristics as interactions, not necessarily as system units.

- **Plug compatibility.** All system units are individuals. Plug compatibility means common interface and interaction standards. Applying pattern ▶"Universal member" provides a common interface through shared skills. Common interaction standards exist where a groupware provides equal access to all participants and connects them reliably. This is quite trivial for groupware based on ▶"Agile message" as participants just need a mobile phone.

- **Non-hierarchical interaction.** Members communicate directly; there is no dedicated organizer for tasks and appointments. Even multi-party and group communication does not need central organization because ▶"Agile message" has broadcast delivery as its default. ▶"Agile message" satisfies the emphasis on interaction in agile design: it barely constrains unmediated interpersonal interaction because it offers many affordances, including context-rich informal communication. ▶"Agile message" could be compared to "voice paper", offering as many affordances for voice messaging as there are for written notes in paper-based collaborative work.

- **Deferred commitment.** The agile paradigm advises transient relationships: deferring commitment until necessary and loosening it the minute it becomes unnecessary. The linear organizational paradigm, on the contrary, imposes permanent commitment after the up-front planning effort. ▶"Agile message" follows the agile paradigm: it offers no feature to enter detailed up-front planning but supports reliable communication, to negotiate and re-negotiate all commitments on demand. In linear "heavy-weight" organization, permanent commitments emerge also from the laborious obligation to maintain comprehensive documentation of knowledge and activity. Again, ▶"Agile message" follows the agile approach: information is stored in people's heads by default and accessed on demand by the reliable means of communication which ▶"Agile message" provides. For example, tasks might be represented by a mnemonic title only.

- **Distributed control and information.** In agile organization, "units respond to objectives" (▶object 7, p. 26), whereas in linear organization, units respond to the steps of a fixed plan ▶[Howell et al :a01, p. 3]. ▶"Agile message" allows individuals to communicate

objectives and therefore it is compatible with agile collaboration. The key idea of ▶"Agile message" is to avoid centralized control of coordination in the CSCW tool, and to distribute this control to the individuals instead. Distributed control implies that "decisions [are] made at point of knowledge" (cf. ▶object 7, p. 26). For coordination decisions, a CSCW application by itself is "a point of no knowledge" while individuals are "points of partial knowledge" due to their awareness and experience. So individuals can reach the point of knowledge with the least effort, which means that agile design principles render it reasonable to distribute control to individuals. Lastly, distributed information implies that "data [is] retained locally but accessible globally" (cf. ▶object 7, p. 26). ▶"Agile message" sticks to this idea: individuals are not forced to document their activity and decisions in a central place, but global accessibility is supported by reliable message-based communication. From the agile organizational viewpoint, three mnemonic keywords are perfectly valid as a task representation, if one can request a longer description where a demand for clarification arises.

- **Self organizing relationships.** ▶"Agile message" can be combined with the ▶"Subscription" pattern, which allows individuals to create and join communicating groups. In terms of the agile design principles, these groups are "dynamic unit alliances" (▶object 7, p. 26), which is a variety of self organizing relationships.
- **Evolving standards.** This principle advocates an "[e]volving open system framework capable of accommodating legacy, common, or completely new units" (▶object 7, p. 26). The system units are individuals. Then, "legacy units" are those individuals who want to stick to their legacy style of collaboration instead of using the automation offered by ▶"Agile message"[30]. Unmediated communication affords collaboration between individuals, independent of the coordination tools they use. ▶"Agile message" affords this just as well, as it constrains communication as little as

[30]This is in no way meant derogatory to those individuals being reluctant to use the new technology. The term "legacy unit" is just a consequence of applying terms of systems engineering to a sociotechnical collaborating system. Admittedly, this perspective is extremely abstract; but it has proven very handy up to now.

possible for message-based communication. In this context, evolving standards are the conventions which emerge in communication.

Appropriateness for community life. Groupware based on the ▶"Agile message" pattern can be used as just a communication support tool, because using the automated artifact interactions is not obligatory. This affords simplicity of usage for those members of the community who are unfamiliar to CSCW technology yet. It also affords collaboration with members who adhere to their legacy methods of coordination.

Audio input in combination with the ▶"Conversation paradigm" pattern is apt for the conversational style of communication desired by communities: it is expressive, informal, enjoyable and socializing. It provides a means for socializing communication throughout the day, advancing community coherence. For these reasons it seems justifiable to expect delightful integration into community life for CGW, at least as a communication-support tool.

The integrated nature of communication and coordination, as provided by ▶"Agile message", seems to be a key to voluntary acceptance by communities. Formal work-style coordination would be rejected; it is difficult even to imagine a group of teen friends filling WAP forms to coordinate a barbecue party in the evening.

Related patterns.

- ▶**Conversational message.** The first pattern combined into ▶"Agile message". It supports agile communication in distributed settings.
- ▶**Interacting artifact.** The other pattern combined into ▶"Agile message". It supports coordination by automated communicative acts and provides ▶"Agile message" with a clearly defined system structure.

4.2.4 Connectable artifact

Software applications offer unintegrated partial models of reality while reality offers seamless integration. As a solution, it is proposed to handle model elements generically, using a common interface called "connectable artifact". This interface resembles the character of real-world things and their connections.

Problem. There are many software applications which model aspects of the real world. Examples include CSCW and CSCL applications, content and document management systems, e-mail and instant messaging applications. But while reality is one big integrated system, these partial models are isolated. What could be an adequate foundational concept to integrate different partial models generically?

Context. In CSCW context, it is no new insight that an integrated collaboration environment is far superior to a set of unintegrated tools (cf. ▶p. 29 on the environment and tool paradigms). It is however no trivial task to find foundational concepts for structuring a collaboration environment, esp. for agile collaboration. An additional constraint is to find a remarkably simple solution because the ▶"Connectable artifact" pattern is intended especially for the community context (cf. ▶p. 34).

Solution. Use connectable artifacts as the foundational concept. Artifacts are separable, identifiable and significant entities which users can create, modify and delete. Being connectable means that users can relate artifacts to other artifacts by different types of relationships. Each connection has a direction and a relationship type.

If one introduces artifacts with different qualities and behavior, they should be of different type, as shown in Xu's CSCL artifact model (cf. ▶p. 29)[31]. For example, there could be unstructured media-specific artifacts for communication, structured artifacts for information storage, and others[32].

Evidence: Rationale. First of all, this pattern does indeed implement the environment paradigm: the limitations of the tool paradigm are overcome, not by removing the distinct tools but by handling their products alike, as

[31] Or see his doctoral thesis in original ▶[Xu :a01, pp. 111–116$_{+12}$]; it is the work that inspired the ▶"Connectable artifact" and the depending patterns.
[32] If there are enough artifact types to express every form of collaboration, a unified "collaboration medium" could emerge from this idea.

"artifacts". The "artifact" concept resembles the character of real things and their relationships; it is simple and intuitive to understand, so that groupware which is based on it will be attractive for community.

An agile framework provides support for agile interactions ▶[Dove :a04, p. 9.5$_{-9.2}$], but cannot guarantee that these interactions are used to build agile systems. ▶"Connectable artifact" can be used for agile interaction of units in an agile system because it is compatible with the agile design principles that deal with interactions (cf. ▶object 7, p. 26), as follows:

Artifacts and their connections add structure and navigability to communication by separating it into distinct entities but do not restrict ways and contents of communication. This unrestricted communication between system units can serve both to share information and to exercise control. Thus, all agile design principles which deal with interaction are implementable with artifacts, among them *non-hierarchical interaction* and *distributed control and information*.

A summary. Object-oriented programming was a laboratory to develop the agile design principles ▶[Dove :a04, p. 9.14$_{-9.2}$]. In object-oriented systems, object exchange is a means of communication; the object has the character of a reified interaction. In analogy, artifact exchange affords communication of system units, and this kind of communication is as unrestricted as exchanging objects in object-oriented systems.

Related patterns.

- ▶**Activity set.** A pattern that can operate on the common "artifact" interface which is introduced by ▶"Connectable artifact" to overcome the limitations of unintegrated CSCW tools. ▶"Activity set" can be implemented as a type of artifact.
- ▶**Agile collection.** Another pattern that can operate on the common "artifact" interface. ▶"Agile collection" can be implemented as a type of artifact.
- ▶**Interacting artifact.** This pattern extends the concept of artifact connection in ▶"Connectable artifact", to include artifact-specific behavior into a generic artifact management application.

4.2.5 Conversational message

Immediate communication is very helpful for the initiator but obtrusive for the targeted individual. This can be balanced by introducing message-based conversation, which exhibits a near-synchronous character. When using voice input, message-based conversation is fast and convenient even with mobile devices.

Alias. Intermittent dialog medium

Problem. Agile collaboration emphasizes "direct negotiation, communication, and interaction" ▶[Dove :a04, p. 9.6$_{-9.2}$]. This is entirely feasible for fully-committed collaboration in co-located settings, for example extreme collaboration as presented in ▶[Mark :a01]. But for distributed collaboration, the obtrusiveness is too high; people might be busy with totally unrelated tasks. How to reduce obtrusiveness while retaining the agile character of direct communication?

Context. Distributed communities are settings where only a fraction of people's time is dedicated to the community. Collaboration will be interwoven with daily routine and private life which allows only a little obtrusiveness (cf. ▶p. 36, ▶p. 42). Additional constraints on a medium for agile and mobile communication are communities' preferences: expressive communication (cf. ▶p. 40); simplicity of technology (cf. ▶p. 34); communication for enjoyment (cf. ▶p. 39); communication for socializing purposes (cf. ▶p. 39); low price of technology (cf. ▶p. 36); and the integration of legacy technologies (cf. ▶p. 36)

Solution. "Tunnel" a conversation over an audio message-based medium. Each contribution to the conversation is an audio message. It is announced by a non-obtrusive token and can be heard by the recipient when time permits it. This is an implementation of both ▶"Conversation paradigm" and ▶"Message". Non-trivial aspects are implemented as follows:

- *Leaving opportunity* to speak (cf. ▶p. 84) as allowed by ▶"Conversation paradigm" is implemented by ending a message.
- *Reliable delivery* (cf. ▶p. 83) is implemented by pull-based delivery of the actual audio message. There should be no possibility to save

the audio message locally, as this makes automatic and immediate acknowledgments of receipt possible.

- *Non-committal overhearing* (cf. ▶p. 84) is implemented by receiving messages immediately while one does not need to reply to them immediately.

- *Answering before receiving the queue impossible* (cf. ▶p. 84) is implemented by the electronic counterpart of "hand-raising" as known from school: the systems grants the possibility to speak not until the current speaker finished.

- *Unarchived* (cf. ▶p. 84). Messages are not archived for access by other people, but they are stored to possibly re-access attended conversations later. This non-ephemeral-character will probably facilitate coordination.

- *Interrupting each other possible* (cf. ▶p. 84) cannot be implemented as the ▶"Message" pattern implies that messages are atomic, once created.

Evidence: Rationale. This pattern is a combination of ▶"Conversation paradigm" and ▶"Message" (cf. them for details). The former provides an agile, context rich style of communication which avoids many problems inherent in current CSCW tools. The latter provides unobtrusive communication.

Audio input is a fast input technique for rapid and spontaneous interaction — it therefore fits the agile paradigm. Audio will be preferred by communities also because it is more expressive than text and therefore allows a more playful and humorous usage. Audio can be sufficient to provide a usable and sociable media space ▶[Ackerman et al :a01, pp. 244-245$_{-237}$]; but of course this is just a hint as it does not necessarily extend to this message-oriented variant of a media space. Communities will welcome mobile audio instant messaging as a means to stay in touch with each other (cf. p. 39), far more reliable than instant messaging from desktop PCs which is currently popular for this affordance ▶[Nardi et al :a01, pp. 84-85$_{-78}$]. At least it is quite obvious that a coordination-only tool which does not integrate free-form communication would be rejected by communities from the very beginning.

Also, audio will be preferred by communities for the convenience and simplicity of input, especially when using mobile devices. Even with instant messaging at desktop PCs, people prefer to talk (by phone) over typing in cases where discussing matters by instant messaging needs long answers ▶[Nardi et al :a01, p. 86$_{-78}$].

Audio integrates seamlessly with the capabilities of mobile phones; they get integrated as a legacy technology people are used to. No up-front investment of money is required and, with flat price models, communication volume is not governed by monetary considerations either. Voice-based instant messaging even reduces the obtrusiveness and annoyance of the mobile phone as argued by Donald A. Norman in ▶[Norman :a05]. All of this fosters voluntary adoption.

Related patterns.

- ▶**Interacting artifact.** A pattern combinable with ▶"Conversational message" because both artifacts and messages exhibit properties comparable to physical objects. ▶"Interacting artifact" supports coordination by automation of communicative acts.
- ▶**Agile message.** A pattern which combines ▶"Conversational message" and ▶"Interacting artifact".
- ▶**Conversation paradigm.** One of the two patterns combined into ▶"Conversational message".
- ▶**Message.** The other pattern combined into ▶"Conversational message".

4.2.6 Conversation paradigm

Text-based CSCW suffers from communication problems unknown in face-to-face conversation. As a solution, conversational style is resembled. Copresence is however not enforced as it can be too obtrusive at times and is unnecessary.

Problem. Traditional text-based CSCW applications suffer from a complex communication problem as they lack proper support for the following points:

- **Awareness.** The tools do not create awareness of the current situation and the social context (cf. ▶p. 29). This is a severe neglect

as the doubling of performance resulting from radical co-location implies "that if we are to truly support remote teams, we should provide constant awareness and easy transitions in and out of spontaneous meetings." ▶[Covi et al :a01, p. 339$_{-338}$].

- **Group coherence, shared experience and trust.** CSCW applications lack support for these, which arises just as the lack of awareness from unbounded, uncertain connections or high turnover participation ▶[Adler et al :a01 p.210$_{-209}$].
- **Non-verbal communications.** This makes up for approx. 70% of communicated information. So text-based communication tools such as e-mails and documents are confined to the 30% verbal information ▶[Aguanno et al :a01, p. 42$_{-14}$]▶[33].
- **Low-latency communication.** Extreme collaboration minimizes all latency times, resulting in greatly reduced end-to-end times of projects ▶[Chachere et al :a01, pp. 3-4$_{+2}$]. This is not possible with current tools such as e-mail, forums and instant messaging. Here, real-time communication is impossible: one cannot guarantee a maximum latency for reading or answering a message, much less can one guarantee the minimum latency which is achievable.
- **Low-overhead information exchange.** The agile paradigm promotes direct interaction (cf. ▶object 7, p. 26) to get rid of the overhead implied in hierarchical interaction, comprehensive documents, plans and contracts (cf. ▶object 6, p. 23).

Context. This pattern helps greatly in agile collaboration settings as the concerned communication problem is fatal here: agility values "[i]ndividuals and interactions over processes and tools" ▶[Beck et al :a03] and therefore strongly advises face-to-face communication ▶[Aguanno et al :a01, p. 42$_{-14}$]. Where this is impossible, as on many occasions in community life, agile collaboration needs CSCW support for rich interaction.

Another context where this pattern fits perfectly is community: here, rich interaction is demanded by the emphasis on socializing communication (cf.

[33]This is however just a rough estimation; Kevin Aguanno admits that mediated communication allows to transport at least some nonverbal information, e.g. in phone conferences where the participants know each other's working style and personalities ▶[Aguanno et al :a01, p. 42$_{-14}$].

▶p. 39), expressiveness (cf. ▶p. 40) and fun (cf. ▶p. 39). Beyond this, ▶"Conversation paradigm" can be implemented so that it matches additional constraints of community life: the groupware must exhibit simplicity (cf. ▶p. 34), be mobile (see pattern ▶"Unbound communicator"), use available technology (cf. ▶p. 36) and heed the costs of communication (cf. ▶p. 36).

Solution. Re-structure communication according to the paradigm of spoken conversation:

- **Continuous opportunity.** Users need not to establish connections to contribute but they do so spontaneously and informally, i.e. without the need to stick to greetings, explicit context establishment or other formal requirements.
- **Contextualized contributions.** Contributions can be lightweight, by making their comprehension depend on previous contributions.
- **Serialized contributions.** All contributions are in linear order. Creating tree-like threads is impossible.
- **Open.** For new participants.
- **Reliable delivery.** The transmission is reliable enough for sender and recipient to trust in it without explicit acknowledgment. This implies push-based delivery; the alternative is unreliable because actions of the recipient are necessary for successful delivery.
- **Real-time communication.** In a face-to-face conversation, answers can be expected in due time. This is possible because face-to-face conversation is synchronous communication.
- **Broadcast communication.** By default, a contribution is delivered to all participants, even if relevant for many of them as awareness information only. To address individuals or sub-groups only, special action is needed, in analogy to sub-grouping or whispering in spoken conversation.
- **Creation-order receipt.** It is impossible to receive contributions out-of-order.

- **Answering before receiving the queue impossible.** In unmediated spoken conversation one listens until one answers. In message-based communication one has to receive the queue of messages before one answers. The queue includes messages under creation.[34]
- **Participating before receiving possible.** One can join a conversation and immediately utter an out-of-context message.
- **Interrupting each other possible.**
- **Leaving opportunity.** By ending a contribution one leaves the opportunity to speak to others.
- **Overhearing makes addressable.** Overhearing a conversation makes oneself addressable to the speakers.
- **Non-committal overhearing.** Overhearing a dialog imposes no responsibilities, not even to acknowledge received content. This is true even if the content is personally relevant — until one is directly addressed.
- **Social norms govern.** Access rights, privacy and confidentiality are managed and enforced by social means, not technically.
- **Unarchived.** By default, conversations are not recorded to be accessible by non-recipients. However, tools might support the recipients' memory by a message log, as it is done in instant messengers.

Evidence: Rationale. This pattern's problem is unknown in face-to-face encounters in agile co-located collaboration (as explained in the problem description). For this reason, surrogating face-to-face communication and copresence by technology has been proposed as the solution to this pattern's problem. Media spaces and MUDs have been used to do so, "creating persistent, predictable, multi-user connections that support a wide range of user interaction and collaborative activity." [Adler et al :a01 p.210-209].

[34]Instant messaging is a counterexample here: due to race conditions, messages under creation are not necessarily received before sending one's own message. This can decontextualize the skipped message, e.g. by moving to another topic.

But copresence is not just a solution, it also introduces new problems. This can be derived from the fact that people prefer solitary work and being alone on many occasions. This applies to media spaces just as well: in the Thunderwire audio space, people were connected less than half of their working time on average ▶[Ackerman et al :a01, p. 241$_{-237}$]. In copresent settings, being approached for synchronous communication can be obtrusive and people need means to signal unwanted or impossible interaction in socially acceptable ways; again, this applies to media spaces ▶[Ackerman et al :a01, p. 244$_{-237}$].

These issues with copresence (whether by physical co-location or in media spaces) seem to be outweighed by the benefits in work settings: the success of radical co-located work ▶[Chachere et al :a01], ▶[Covi et al :a01], ▶[Mark :a01] and long-term media space usage ▶[Adler et al :a02] points in that direction. And even in domestic settings, media space usage in analogy to office-shares is promising ▶[Bayley et al :a01, pp.328-329$_{-324}$].

There are however settings where the drawbacks of copresence will outweigh the benefits, for example mobile applications that are not limited to the work hours (cf. this pattern's context). Where copresence is provided through a mobile application the individual has literally no space left to be on his own; the perceived obtrusiveness will probably be that high that people tend to be logged off whenever they want to be undisturbed. This however will render the media space useless for solving this pattern's problem: neither awareness nor group-coherence, shared experience or low-latency communication could be provided; a media space does not provide reliable communication where people tend to use it on a connection basis. Such usage patterns endanger even voluntary adoption as the critical mass might not be reached (cf. ▶p. 44).

These considerations suggest not to propose a media space as the solution but to remove copresence if possible and provide just the principles by which a media space solves this pattern's problem. These principles are shown in the solution. They model how spoken dialogs work; that way they avoid yet unknown problem-laden affordances of "invented" communication

principles[35] and at the same time address community's emphasis on simplicity (cf. ▶p. 34).

While the ▶"Conversation paradigm" pattern is innovative in its comprehensiveness, it can be justified from existing CSCW tools at least partially. The following paragraphs show how subsets of this pattern's principles solve subsets of its problem. These experiences have been made with various types of CSCW tools.

- **Awareness.** Instant messaging, which resembles conversational style in many ways, provides awareness information within the stream of messages:

  > Conversations can be more interactive because the rapid and evolving nature of IM means that there is immediate context for the current interaction. ▶[Nardi et al :a01, p. 81$_{-78}$]

  So *non-committal overhearing* of group communication without active participation is a worthy practice: it provides awareness of what is going on. All of this does not need the obtrusiveness of copresence: instant messaging gets around this by delayed responding and plausible deniability of presence ▶[Nardi et al :a01, p. 84$_{-78}$].

- **Group coherence, shared experience and trust.** Group coherence and trust are social qualities. To provide them, the medium must afford socializing communication. Again, instant messaging can serve as an example. It is supposed "that a key reason for this informality lies in the near-synchronous nature of IM. Conversations can be more interactive because the rapid and evolving nature of IM means that there is immediate context for the current interaction. This context seems to reduce misunderstandings and promote humor." ▶[Nardi et al :a01, p. 81$_{-78}$]. Now humor is a socializing form of communication and it is important in community context. The example of instant messaging suggests that a medium provides social qualities like group coherence and trust not by synchronous or copresent communication but by a conversational communication style.

[35]For example, inventing that all spoken dialogs are archived by the media space will rise issues with privacy, confidentiality and access rights.

High-turnover participation hinders group coherence and shared experience ▶[Adler et al :a01 p.210$_{-209}$], and dyadic communication does the same. So facilitated *broadcast communication* would be a solution. ▶"Conversation paradigm" applies this idea: it makes broadcasting the default[36] and enforces neither copresence nor synchrony, which makes group conferences much easier to establish.

- **Non-verbal communications.** Conveying non-verbal content works best with synchronous media ▶[Pankoke-Babatz :a01, p. 14$_{+16}$] but is not limited to them. Near-synchronous communication like instant messaging is found capable of that also ▶[Nardi et al :a01, p. 81$_{-78}$]. ▶"Conversation paradigm" implies near-synchronous resp. quasi-synchronous communication, in the sense that messages are exchanged in a synchronized way without concurrency problems. Therefore it will hopefully be found that it can transport non-verbal content.

- **Low-latency communication.** While instant messaging does not guarantee low latency, its latency is low on average ▶[Nardi et al :a01, p. 81$_{-78}$]. This is due to the *continuous opportunity* to communicate that it provides, resembling informal workplace conversation ▶[Nardi et al :a01, p. 82$_{-78}$]. By enforcing the *real-time communication* quality of the conversational paradigm it seems therefore possible to achieve guaranteed low latency without copresence.

- **Low-overhead information exchange.** It is a concept at the heart of agile software development to communicate rather than to write documents (cf. ▶object 6, p. 23). In a groupware, the lumbering overhead should be defined as anything unnecessary to accomplish the current task, including comprehensive and beautiful project and task documentation in CSCW tools if one can do without it. The least overhead is implied in informal on-demand interpersonal communication. It spares the overhead of materializing memorized information that never gets used.

  Again, instant messaging illustrates that co-presence is not

[36] This addressing style is simple and effective: for this reason, mailinglists facilitate group communication while e-mail per se does not.

necessary here ▶[Nardi et al :a01, p. 82–78]. Agility concentrates on this style of interaction also: in many cases it is enough that conventions and *social norms govern*. Thereby, the inflexibility and overhead is avoided which arises from dealing with explicitly typed entities and the application features dedicated to them. One example is calendar usage in domestic settings: access rights, coordination protocols etc. are all implemented in social discourse, whose informality provides a maximum of flexibility; cf. ▶[Crabtree et al :a11].

Related patterns.

- ▶**Message.** Because ▶"Conversation paradigm" does not enforce synchronous communication it is combinable with message-based communication.
- ▶**Conversational message.** A combining pattern of ▶"Conversation paradigm" and ▶"Message", implementing an asynchronous media space.

4.2.7 Interacting artifact

How to implement artifact-specific features into a generic artifact-management application? This pattern proposes to tie behavior to the artifacts, remove artifact types and to allow unrestricted interactions between artifacts. This solution is then applied to artifact-based, agile groupware.

Alias. Reified interaction[37]

Problem. For the task of managing artifacts which are not machine-understandable, applications are often structured as specialized tools. Such a tool exhibits reasonable behavior because such behavior was implemented into its features and because it collects machine-understandable meta-information about the artifacts.

For example, image management software might offer features like creating albums. It will store the containment relationships between albums and

[37]This name resembles the view that groupware artifacts like tasks, appointments, messages etc. are communicative acts, i.e. interactions between individuals. Confer the introductory notes on the MC³ (▶p. 60).

images in a central database. This meta-information is unaccessible from a generic artifact management software, say, a file manager.

An artifact-based application is a superior alternative (cf. pattern ▶"Connectable artifact"). It is a generic application, integrating artifacts of arbitrary types. Which implies that it must handle all types by their common "artifact" interface. Having no knowledge about the internals of each type, it cannot offer specialized features or collect type-dependent meta-information. How is meaningful behavior possible with such an application?

Context. The pattern is developed with the CSCW context in mind[38], so that a more specific verbalization of the problem can be put like this: How can an artifact-based CSCW application provide automation of coordination? Also, the pattern is to be applied especially in community context. It must therefore meet community's preference for simplicity of technology (cf. ▶p. 34).

Solution.

Distributed control and information. Distribute machine-understandable information to artifacts by annotating them with meta-information; and distribute control to artifacts by annotating them with behavior[39]. Doing so allows generic artifact management and artifact type specific functionality at the same time. Each kind of annotating behavior is an interaction. Interactions are in general possible between all system units[40].

Typeless artifacts. To view artifacts, they would be annotated with a behavior how to be viewed. If however artifact types differ in allowed interactions only, not in internal structure, viewing artifacts can be implemented into the application instead of in modularized behavior. Where the interactions of different artifact types have central aspects in common, it might even be possible to go a step further and allow every combination

[38]Examples of CSCW artifacts which are not machine-understandable include: text, video and audio artifacts containing tasks, memos, proposals, survey data etc. in natural language.
[39]In CSCW applications artifacts might e.g. contain text, audio or video data. There are two alternatives if annotation is done upon creation: the annotating content might be repeated within the artifact content or not. The former imposes redundancy but no formally constrained conversation, the latter has it vice versa.
[40]Connections (see pattern "Connectable artifact") do not constrain possible interactions either; they are no interaction-constraining system structure but rather "time-invariant interactions".

of artifact interactions for all artifact types, effectively removing the “artifact type” concept.

Modularized interactions. Artifact interactions, i.e. new functionality, can be added to the application as separate modules. Such modules are not separate tools, they integrate tightly with each other. Users might develop their own specialized interactions.

Suggested interactions for agile groupware. Here is a suggested set of primitives for coordinating interaction, applicable to typeless artifacts:

- **Root artifact.** Only one artifact can have this interaction set. It acts as entry point to the space of connected artifacts. It could be presented as an application’s “desktop”, for example.
- **Contribute artifact to group conversation.** A combination with patterns ▶“Conversational message” and ▶“Negotiated synchrony” is profitable here. Then, the interaction takes the “latest time of delivery” as a parameter, and reminders are sent to the participants to make them aware of the new contribution to conversation. Artifacts which are not contributed to group conversation or linked to publicly accessible artifacts are only accessible by their creator. This allows integrated management of personal and group information.
- **Show as list entry of target artifact.** Artifacts targeted by these dyadic interactions provide a list of artifacts. There are multiple kinds of lists, differing in the way they order items: by creation date, in alphabetical order or according to parameters which denote a preceding artifact for each artifact. Effectively, these lists act as coordinating representations, e.g. as group task pool, personal task box, note box, to-do list, buying list, or even as a menu for other coordinating representations.
- **Show as calendar entry of target artifact.** Artifacts targeted by this dyadic interaction provide a calendar, i.e. act as time-coordinating representation. This interaction takes a date or date range as parameter.
- **Include in overview.** All artifacts affected by this interaction are put into a graphical overview which provides awareness of upcoming

tasks, appointments etc. at a glance. It is a powerful utility for coordination decisions. Access is possible from a web interface or through regular delivery to registered e-mail addresses.

- **Remind the artifact.** Reminders are broadcasted to the whole group by default but recipients can be limited manually. There are multiple reminder types, each of them takes a reminded artifact as parameter:
  - **Time-based reminder.** Start and end time of the reminder's activity are adjustable.
  - **Location-based reminder.** For inspirations cf. ▶[Chan et al :a01].
  - **Event-based reminder.** For events of which the groupware is aware, e.g. the deletion of the artifact or the moment a certain person joins group conversation.
  - **Materialized reminder.** A materialized reminder is a sticker which contains the following information about an artifact: an identifier; a text title; and a pictogram hinting at the content's character (warning, to-do item, instructions etc.). Placing such stickers on objects can integrate explicit guidance into the task environment, which supports pragmatic action ▶[Alterman et al :a10, p. 53–52]. The reminded artifact might be e.g. a coordinating representation to manage the objects availability or necessary maintenance work.
  - **Inviting reminder.** The reminder's behavior is tuned to resemble invitations to upcoming meetings.
  - **Countdown reminder.** Probably, this one will be used semi-seriously out of a sense of pleasant anticipation.
- **Block interactions.** This renders an artifact inactive. Just as obsolete task notes on paper are thrown away, artifacts are coordination utilities for temporal use, not to document activity formally. One can also limit the blocking to a time span.
- **Distribute as awareness information.** Distributing awareness information can be integrated into the flow conversation, as done in conversations unmediated by technology. However, the ephemeral

character of verbal communication introduces uncertainty which imposes extra work. Rogers and Brignull discuss this problem in ▶[Brignull et al :a01], arguing for a visual externalization of awareness information. Where this is impossible, e.g. when limited to audio artifacts, a near-equivalent might be provided by a special form of distributing awareness information. All audio artifacts marked as awareness information are not ephemeral but inserted into a collection of artifacts which participants can access at any time, e.g. to check the current situation before starting a new task. The collection is sent to people upon entering the conversation. The artifacts in this collection might be annotated with an expiry date or deleted if a new artifact from the same participant is inserted.

- **Collect information.** This interaction makes a contribution to group conversation, asking for certain pieces of information. The question can be entered in natural language and might e.g. request registration of event attendees, comments on possible optimizations, instructions, experiences, opinions to aid a decision process, ideas like in brainstorming, or awareness information. The collected information is stored into artifacts and linked to the question. This interaction might be triggered immediately, at a specified point of time or by inactivating an artifact. One can choose whether every addressee must or may answer within a given time, or without time limit.

Evidence: Rationale.

Arguing for distributed control and information. Whenever artifacts are not machine-understandable, an application to handle them will have specialized features and will collect machine-understandable management information about these artifacts. Such an application is a specialized tool with centralized control and information. Generic artifact management is possible only with self contained artifacts, i.e. artifacts which have information and control distributed to them.[41]

[41]There is a casual coincidence with the agile design principle “distributed control and information” (▶object 7, p. 26) here. Agile collaboration of individuals does however not demand that the artifacts used for communication are managed in agile manner, too.

Arguing for typeless artifacts. The key idea here is to provide basic concepts with many affordances instead of a realistic model of reality. In a groupware, these concepts can be employed in agile and unanticipated ways to coordinate and communicate, governed by conventions. A typeless artifact is a multi-purpose concept whose purpose can be expressed by attached interactions. This idea is true to the practices of agile software development where individuals and interactions are valued high but comprehensive documentation is valued low (cf. ▶object 6, p. 23).

Xu introduced the idea of artifact-based groupware; in his artifact model, there are different types of information and coordination artifacts; the user can configure which application is invoked when accessing an artifact of a specific type ▶[Xu :a01, pp. 111-112$_{+12}$]. In effect his artifact model is a meta layer which integrates specialized tools by connecting them to the artifact types they can handle. This is no full integration as functionality from different tools cannot be combined and artifacts cannot interact with artifacts of an arbitrary type. Removing the "artifact type" concept serves much tighter integration.

Removing artifact type is not possible in all contexts but seems possible in the groupware context. Groupwares tend to make the user think that appointments, tasks etc. are "in" the groupware. However, the groupware artifacts are separate from these real-world entities; they are just there to communicate about these real-world entities. There is no reason to create a realistic model of reality in communication. For example, where notes on paper are used for coordination, they do not differ by "type" but can be handled all the same way, according to their basic physical properties as some text on paper (cf. also ▶[Crabtree et al :a11, pp. 127-128$_{-118}$]). Thus, notes on paper can take different roles in an agile way. For example, it might be created as a shopping list, then it might be a message given to the person who does the shopping, then serve as utility to calculate each party's expenses and lastly be archived, with some additional notes. To conclude, it seems enough to provide a generic coordination artifact for CSCW applications and to allow unrestricted interactions between artifacts.

Removing the "artifact type" concept this way provides a simplicity and flexibility which will hopefully make the groupware attractive for communities. It resembles the informal style of communication in

communities, which is not formally conceptualized in terms like “task” and “appointment”.

Arguing for modularized interactions. This expands the framework of possible interactions between individuals in computer-mediated agile collaboration. The groupware can be adapted to new situations, new types of tasks and skills. Thus, adding new interaction possibilities through modules implements the agile design principles “evolving standards” (cf. ▶object 7, p. 26), which deals with continuous development of the framework.

Arguing for the suggested interactions for agile groupware. Unrestricted interactions of typeless artifacts allow flexible, creative uses. But to justify the interaction idea and the proposed interactions, it is enough if the typical uses of current groupwares are possible. One typical use is task management. Recreate it as follows: create an artifact. Then, use the interaction ▶“show as list entry of target artifact” and a target artifact which has, by convention, the role of a task pool or to-do list. If the task’s end date is important, use one of the possibilities of ▶“remind the artifact”. Optionally, use ▶“contribute artifact to group conversation” to make the group aware of it. If the task is done, use the interaction ▶“block interactions”. Managing appointments is like managing tasks, only that the interactions ▶“show as calendar entry of target artifact” and ▶“inviting reminder” are used.

Related patterns.

- ▶**Connectable artifact.** Artifact connections are, from the viewpoint of systems engineering, time-invariant interactions. ▶“Interacting artifact” extends this pattern to include dynamic interactions. These interactions contain “latent” connections which will be explicated under certain conditions.
- ▶**Conversational message.** ▶“Interacting artifact” supports coordination through automated artifact interactions. But coordination remains a human task, which means that it is unnecessary to describe machine-understandably what to coordinate. But coordination by humans needs communication support, and this is provided by ▶“Conversational message”.

- **Agile message.** A pattern combining "Interacting artifact" and "Conversational message".

4.2.8 Message

Synchronous communication is obtrusive, especially if people are in reach anywhere, anytime. To solve this, make asynchronous message-based communication the default.

Problem. Telecommunication and data networks empower people to communicate and coordinate anywhere, anytime. But telecommunication devices can interfere with the user's current activity whenever it demands attention or action from the user. Results are distraction, annoyance, obtrusiveness and obligations[42]. Turning the devices off does not solve the problem but removes it together with the benefits.

Context. Mobile devices provide the anywhere, anytime character of communication in particular high degree. For this reason, the mobile phone has gained a bad reputation for its annoyance and obtrusiveness [Norman :a05].

Solution. Do not enforce communication to be synchronous. Instead, split the flow of communication into individual chunks that can be sent immediately but are delivered at the recipient's choice, when setting and workload considerations permit it. Users might switch to synchronous communication when needed.

Evidence: Rationale. Synchronous communication means immediate delivery. The time of delivery then depends solely on the sender's action and thus cannot be anticipated by the recipient. To remove the potential of bothering or interrupting the recipient, the sender needs to be aware of the recipient's state. Initiating dialogs works that way if people are co-located.

This solution can be automated; which relieves the sender from memorizing his message and monitoring awareness information, and it relieves the recipient from providing awareness information. The "Message" pattern is this automated solution.

[42]Obligations come partially from social forces: people report that they prefer instant messaging over phone communication to avoid lengthy conversations when they need just a quick answer [Nardi et al :a01, p. 81-78].

Related patterns.

- ▶**Conversation paradigm.** In combination with this pattern, message-based communication acquires important characteristics of face-to-face conversation. This solves some problems with mediated communication.
- ▶**Conversational message.** Combines ▶"Message" and ▶"Conversation paradigm". As a side-effect, message-based communication brings structure and navigability to conversation. This can be utilized to integrate conversational messages as artifacts into groupware applications.
- ▶**Negotiated synchrony.** As a trade-off from applying the ▶"Message" pattern, latency times become incalculable. This effect is removed by ▶"Negotiated synchrony". With it, a sender can prevent that a message is delivered when it is already outdated.
- ▶**Subscription.** A helper pattern that fosters mutually relevant communication, regardless if message-based or not.
- ▶**Unbound communicator.** The unobtrusive style of communication provided by ▶"Message" is especially relevant for reliable anywhere, anytime communication as proposed by ▶"Unbound communicator".

4.2.9 Negotiated synchrony

Continuous mobile instant messaging can be obtrusive in spite of its message-based character. This pattern provides a synchronism negotiation protocol to cope with this: recipients may accept only a certain level of synchronism resp. obtrusiveness; senders may override this for urgent messages.

Problem. Communication technology may allow to stay in touch anywhere, anytime by using a variety of instant messaging services on mobile devices. Here, obtrusiveness becomes a problem, especially when people must or want to be undisturbed. How to solve it while retaining the desired characteristic of being in touch with each other and the conversation-like properties of instant messaging?

Context. While this pattern is quite general, it was originally designed for distributed agile collaboration. Here, staying in touch is strongly desired: "if

we are to truly support remote teams, we should provide constant awareness and easy transitions in and out of spontaneous meetings" ▶[Covi et al :a01, p. 339–338]. (This conclusion was drawn after experiments with radical co-located collaboration had shown a doubling of performance.) ▶"Negotiated synchrony" provides a means to initiate and end spontaneous meetings, namely to negotiate a high degree of synchrony for the duration of the meeting.

▶"Negotiated synchrony" relies on socially acceptable behavior because a user's request to be undisturbed can be overridden; this is meaningful for urgent messages but could be exploited. Therefore, this pattern is only appropriate without changes where socially acceptable behavior can be enforced through social forces, e.g. in small closed groups like communities.

Solution. Addressees must provide information on their "obtrusiveness acceptance level" constantly. The value might be one of: "immediate delivery possible", "later delivery favorable", "later delivery requested", "later delivery enforced", maybe annotated with a short reason. Whenever the value is other than its default "immediate delivery", addressees must have a next "sync point" specified; it indicates when they will have caught up with the messages in their queue at latest. These pieces of awareness information are broadcasted to all participants so that they can bide their time for e.g. synchronous communication.

Now senders annotate each message with a "latest time of delivery", defaulting to "anytime". If this time is before the sync point of a recipient, the application asks if the message should be sent anyway, overriding the recipient's sync point. Senders can decide this based on the obtrusiveness acceptance level and their message's urgency.

If however a recipient indicated "later delivery enforced", the latest time of delivery for this recipient is automatically adjusted to this recipient's sync point. This does however not block group communication: the group is simply made aware of the deviating latest time of delivery for this particular recipient. This is effected by an annotation to the message.

Recurring information on the obtrusiveness acceptance level and on sync points can be provided by rules, e.g. for one's work hours. Convention will

govern what should be the maximum distance between sync points. The application will make people aware of upcoming sync points. There should be a "sync now" function which users might use to execute their sync point beforehand and to specify another sync point in the future. If a sync point is reached however, the application uses push functionality to deliver queued messages. Thus, sync points are both reliable for senders and flexible for recipients.

Evidence: Rationale. Fully asynchronous communication (like e-mail) is unobtrusive and flexible: it can be sent any time and received any time. Synchronous communication (like phone calls and to some extent instant messaging) supports conversation-style communication which makes people feel to be "in touch" with each other. This pattern retains all these advantages (however not at the same time) by offering a range of communication synchronism and thus of the necessary obtrusiveness. The value can be adjusted at any time on a per-user basis.

A possible alternative to this pattern is to automate the decision if a message should be sent even though it overrides a recipient's sync point. To achieve this, senders would need to indicate the level of urgency within a message. This is not a good idea however: in most cases the urgency information will go unused, and in the remaining cases it is not sufficient: users would base a decision to override another user's wish for privacy on a thorough consideration of rich awareness information. This cannot be simplified to just a comparison of obtrusiveness acceptance level and urgency level.

Related patterns.

- ▶**Message.** A pattern which implements message-based communication. ▶"Negotiated synchrony" can enhance the intended unobtrusiveness of ▶"Message" by negotiating more asynchrony, or enforce a real-time dialog by negotiating more synchrony. In combination, it provides a fluent and dynamic transition between synchronous and asynchronous communication, just as desired. This effect is estimated very important for agile collaboration ▶[Covi et al :a01, p. 339-338].

4.2.10 Subscription

To cope with information overload and privacy concerns while maintaining openness of communication, provide the possibility to subscribe to groups (or rooms, channels, lists, …) and to dynamically create new ones.

Alias. Subgrouping paradigm

Problem. Information overload is a constant problem in broadcast-style media. Additionally, these media do not allow the discussion of private matters.

Context. Most communities encompass dynamic subgroupings, centered around special interests (cf. ▶p. 33). This pattern can handle subgroup specific communication: it does not overload other members with irrelevant information, infringe privacy or evoke suspicion. It tries to support open subgroupings, in support of the *common* boundaries of community (cf. ▶p. 32).

Solution. Provide a "global" discussion space where all community members are by default. Provide means to support groupings: locating a group, locating people, inviting other people to create or join a group, joining a group, leaving a group, dissolving a group[43]. Joining a group is possible at every time but the other group members get informed of it. Joining multiple groups at the same time is possible. By default, communication is not archived; this makes is feasible to discuss matters which only the current members are allowed to know.

Evidence: Rationale. The difficulties of the problem are its subtle social matters like privacy, openness, suspicion and so on. Handling them should be up to social means, just as in non-mediated communication. Therefore, just provide an environment where social means can be deployed in the form of emerging patterns of technology usage (cf. ▶p. 41). This environment should be comprehensible and flexible; which can be realized e.g. by resembling the important aspects of communication in physical communal places (like a community's meeting room).

Thus the proposed solution is especially applicable to community settings: it resembles multiple groups located nearby in communal settings, where it

[43] A group dissolves if it has two members and one leaves.

is possible to overhear multiple discussions, leave one group to join another, make public announcements and so on.

The underlying basic mechanism is a flexible variant of subscription. Subscription is used widely to reduce information overload and provide open communication at the same time. Well-known examples are newsgroups and mailing-lists. Another example: Gloria Mark relates on a publish-subscribe system that is used as a tool for "extreme collaboration" in NASA's Jet Propulsion Laboratory ▶[Mark :a01, p. 91–86].

And finally, the ▶"Subscription" pattern supports interaction in agile collaboration: it implements the agile design principles "non-hierarchical interaction", "self organizing relationships" and "flexible capacity". And it facilitates the global information access demanded by "distributed control and information" (cf. ▶object 7, p. 26): where information is distributed between collaborating individuals, its access must be facilitated by reliable means to contact any participant at any time. The ▶"Subscription" pattern provides exactly this because all groups can be created and joined anytime.

Related patterns.

- ▶**Message.** A pattern implementing message-based communication. Its targeted unobtrusiveness can be enhanced by ▶"Subscription".

4.2.11 Unbound communicator

A problem of current telecommunication technology is failing connections because people are not where their equipment is. To co-locate equipment and communicator, make the equipment mobile and wearable.

Problem. Telecommunication technology provides independence of co-locating the communicators but depends on co-locating each communicator with stationary telecommunication devices. Thus telecommunication technology is unreliable because of the communicators' mobility: when away from their telecommunication devices, they cannot be contacted and are impeded in contacting others. This applies to synchronous and to a lesser extent to asynchronous communication.

Context. This problem does, of course, affect distributed communities. But it affects all other communities, too: there is no pure co-located community in a mobile world, members are on the road frequently.

Now the urgency and volume of distributed communication in communities is not too high and can be handled by current telecommunication technology. More important is that unreliable communication initiation disfigures the character of community:

> Unbounded, uncertain connections or high turnover participation make it difficult for groups to establish and maintain common awareness, group coherence, shared experience, and trust. ▶[Adler et al :a01, p. 210$_{-209}$]

This is important for distributed communities as they have almost no co-located meetings (the above citation is from a work on network communities). But it is important for more co-located communities also as they have tighter bounds and therefore expect shorter response times.

Additional constraints of community life are the preference for low-cost technology (cf. ▶p. 36) and the demand to respect the latency (cf. ▶p. 36). A key to agile collaboration (like XC) is to reduce latencies to an absolute minimum ▶[Chachere et al :a01, pp. 3–4$_{+2}$]. Though community collaboration does not value efficiency that high (cf. ▶p. 37), something that supports agile, spontaneous collaboration will be welcomed (cf. ▶p. 42).

Solution. Use only those media that are accessible by mobile devices and that can be accessed at an comparable level of convenience from all relevant devices. In communities, do not deploy any communication applications that are not accessible from mobile devices. If (and mostly, because) it is not possible to equip a community with specialized mobile hardware, use the existing mobile phones of the members.

Evidence: Rationale. As telecommunication is impossible without equipment, equipment and communicators must be co-located. Binding the communicator to stationary equipment is not feasible: this pattern would have been long developed if immediate communication would be more important than mobility. So mobile telecommunication equipment must be bound to the communicator. To respect the necessity for low-cost

technology in communities and to respect legacy technology, only mobile phones are used as they are widely in use and comparatively low in cost.

Related patterns.

- ▶**Message.** ▶"Unbound communicator" argues for being accessible for telecommunication anywhere, anytime. As this makes problems with obtrusiveness urgent, ▶"Message" and its helper patterns are needed to solve this.

4.2.12 Universal member

Role-based organization malfunctions where people are not accessible permanently or the inflexibility of being accessible is unbearable. The alternative is to go towards role-free organization.

Problem. Latency, inconvenience and failure arise if people with specific expertise, roles or responsibilities are needed but not accessible. Role-based organization provides meaningful coordination mechanisms by centralizing responsibilities. In change-intensive settings however, people are not guaranteed to be accessible, and the coordination mechanisms executed by them might be unavailable. That way, strict role structure hinders group performance in agile settings.

Context. This pattern applies especially to settings where people's presence and involvement can change in unanticipated ways; so it applies to communities, where involvement and meeting attendance has mostly voluntary character. This pattern can however not be applied economically where tasks demand a high level of expertise: getting rid of roles would here mean that the needed expertise has to be taught with great efforts. But as most tasks in community life can be solved in pragmatic ways (cf. ▶p. 38), this pattern can be applied successfully in this setting.

Solution. Disestablish the roles. To do so, provide organizational and technical means which empower members to perform every group activity at any time. A necessary means is to provide effective communication and coordination tools which compensate for losing roles as "coordination centers". Applying this pattern partially helps a lot but leaves some unreliability.

Evidence: Rationale. This pattern is about organizational design and cannot be directly implemented into agile groupware. But it results from the agile paradigm just as well, as shown in the following paragraphs. The agile design principles (cf. ▶object 7, p. 26) are system generic, i.e. a team of people is a system of interacting units from a systems engineering viewpoint ▶[Dove :a04, p. 9.6$_{-2}$]. Now role-free implementation is found to be compatible with all agile design principles, but most importantly with these:

- **Self contained units.** Agile systems should consist of "separable, self-sufficient units not intimately integrated." ▶[Dove :a04, p. 9.6$_{-2}$]. Roles impose tight integration and lack of independence while universal members are such self contained units.
- **Distributed control and information.** Roles are on the contrary an implementation of centralized control and information. Effective control demands for the co-location of information and decision rights, which might be achieved by moving one of them, depending on the costs of information transmission ▶[Brynjolfsson et al :a01, p. 245$_{-244}$]. Role-based organization advises to centralize both decision rights and relevant information in roles, but information technology empowers to transmit distributed information on demand▶[44]. This is why this pattern contains the suggestion to introduce CSCW tools.
- **Self organizing relationships.** Obviously, a predefined rigid role structure is not self organizing or self adapting. Role-free organization can be self organizing or chaotic. Community, as a space for dynamic subgroupings (cf. ▶p. 33), is already self organizing; it is advisable for communities not to restrict this desirable quality through role-based organization.
- **Unit redundancy.** Role-based organization makes people indispensable: they are not redundant as nobody shares their expertise. Cross-training people on a multitude of tasks introduces flexibility through redundancy and dispenses the role concept. The

[44]Confer ▶[Brynjolfsson et al :a01] for a detailed investigation into the interdependence of modern information technology and the uplift of agility.

PoInter project arrived at a CSCW pattern similar to this one; they call it “Overlapping Responsibilities” and write about its advantages:

> This also builds *redundancy* into the system and provides for cooperation, supervision, advice, sharing of knowledge and so forth as part of the normal group activity.[45] ▶[CSEG :a01]

Rick Dove relates a case study of a stamping plant specialized in after-model-year automobile body parts. Its organization illustrates to the point how agile design principles apply to roles and responsibilities, exactly as captured in this pattern:

> Eight years ago the plant went to a single job classification in production, cross training everyone on everything – a press operator one day might change dies as well, the next day work in the assembly area building hoods in the morning and fenders in the afternoon – and the following day go off to another plant to review a piece of [stamping] equipment or part for how to bring it back. ▶[Dove :a04, p. 9.16$_{-9.2}$]

Summing up: making roles unnecessary through cross-training and CSCW introduces flexibility and reliability. It raises an organizations agility and thus the benefits that can be derived from using agile groupware.

[45] emphasis in original

5 Implementation: a research prototype

A community is like a ship; everyone ought to be prepared to take the helm.

attributed to Henrik Ibsen

Summary. Desiring a first valuation of the MC³ and some feedback from using CGW, a research prototype CGW:RP is implemented here[46]. It implements group communication for MMS with mobile phones.

5.1 Feature set

Summary. This proposes a design of CGW:RP's feature set (a subset of CGW's features). CGW:RP implements a message-based audio space. As this is central for CGW, the prototype shares CGW's character, though it is far from containing all features of the MC³.

Broadcast-style message-based communication. The MC³ has two integrated main features: communication support by message-based audio group conversation and coordination support by interacting artifacts. The prototype focuses on the first aspect; it is the foundational feature as coordination is effected by communication in the MC³. Concretely, the prototype affords group communication by sending combined text and audio messages to a single address. Messages which arrive at this address are automatically broadcasted to all registered group members. In this respect, the behavior of CGW:RP resembles the behavior of a mailing list. Just like a mailing list, CGW:RP ensures that *group* communication takes place: addressing mistakes like omitting group members will not take place, and a recipient can assume that all other group members received the message she received.

Conversation paradigm. CGW:RP includes preliminary support for the conversation paradigm, implementing some central constraints: there is an ongoing conversation offering *continuous opportunity* to participate; contributions are *contextualized* and *serialized*; delivery is *reliable* after an initial test for configuration; contributions are *broadcasted*; contributions are *received*

[46]The full source code of CGW:RP is contained in chapter "B. Source code of CGW:RP" (pp. XXIII).

in creation order; by sending a message, one *leaves the opportunity to speak* to others; overhearing is only possible for the (mutually known) recipients and thus makes *addressable*; *social norms* govern how to deal with messages; and finally, *archived messages* are only accessible by participants.

Memo box. CGW:RP includes basic functionality to store and organize received messages, e.g. to memorize tasks, appointments and others. This feature simply relies on the mobile phone's capability to store and organize MMs. From the MC³ point of view, this is a very limited implementation of the interaction "*show as list entry of target artifact*" (cf. ▶p. 90). As intended by the pattern ▶"Agile message", the short text part of a message helps to organize messages in a memo box while the main content is stored in the audio part (cf. ▶p. 70).

E-mail to MM gateway. While not an intended feature, it is an affordance of CGW:RP which deserves notion: CGW:RP receives e-mails and broadcasts them as MMs, thereby converting some e-mail attachments to MM slide content. This means that CGW:RP can be easily extended to be a general e-mail to MM gateway. Only a few commercial suppliers offered this kind of service in mid 2006[47], so setting it up for personal use could be useful.

5.2 Implementation decisions

Summary. CGW:RP regularly retrieves MMs from an IMAP e-mail account and distributes them via the MM3 HTTP interface of a commercial vendor to the mobile phones of registered recipients. These implementation decisions are presented here together with their justification. In particular it is argued why CGW:RP uses MMS as its basis technology.

The principle of operation. Community members use their mobile phones to send MMs to a single e-mail address. CGW:RP is called every minute by a cron job and checks this IMAP e-mail account for recent messages. It then distributes these messages to registered community members as MMs. For sending MMs, a commercial web service with a simple HTTP interface is employed. CGW:RP needs no web server to operate, calling it from the command line PHP interface is enough.

[47] After extensive web research, only one commercial supplier of this service in Germany was found at end of July 2006: the Whatever Mobile GmbH ▶[Whatever Mobile :a01].

Using MMS messages as agile messages. The central implementation decision for CGW:RP was to use MMs for implementing the ▶"Agile message" pattern. MMS seemed to be the most widespread technology of mobile phones which supports combined text and audio messages natively. These attributes have been decisive for the question which basis technology to use, for the following reasons:

- CGW:RP is intended to gather real-life experiences with CGW in communities. Relying on technology which is currently in use is therefore necessary.
- CGW:RP is a prototype. It will never be used as a final product and therefore should not require too much development effort. This renders it unreasonable to use basis technologies which have no native support for combined text and audio messages, including PTT and SMS messaging combined with phone calls (using a substation number to identify the message to retrieve).

Operating CGW:RP in a group of five individuals means that every MM to CGW:RP entails four MMs to the remaining group members[48]. In daily use, these costs surely would not allow an ongoing group communication. This renders MMS inadequate as basis technology for CGW as a product. But a MMS-based prototype is adequate to gather first experiences within communities, if and only if community members are relieved of the costs of prototype operation.

How to receive MMS in a web application. As of July 2006, it is possible to send MMs from mobile phones to e-mail addresses with all four German mobile network operators[49]. RFC 4356 defines the necessary transformation of header information and specifies that the message body is taken over in unaltered form ▶[The Internet Society :a01, p. 7]. Mobile network operators however perform several steps of content transcoding in addition, e.g. from AMR audio to WAV audio; see for example ▶[T-Mobile :a01]. The commercial MM3 HTTP interface employed by CGW:RP to send MMs demanded to reverse these steps in order to

[48] In Germany in July 2006, this results in 5·0.39 EUR = 1.95 EUR ▶[teltarif.de :a01]. Delivering the MM by a WAP push SMS and subsequent GPRS download results in lower costs if the user has a corresponding GPRS rate (see ▶[teltarif.de :a02] for the German situation).

[49] For T-Mobile, cf. ▶[T-Mobile :a01]; for Vodafone, cf. ▶[Vodafone :a01, p. 2]; for E-Plus, cf. ▶[E-Plus Service :a01, p. 1]; for O_2 Germany, cf. ▶[O2 Germany :a02].

send MMs. To transcode WAV back to AMR, the 3GPP TS 26.104 V 5.2.0 reference implementation of an AMR encoder and decoder is used, which can be found in ▶[3GPP :a04].

CGW:RP uses an e-mail account to receive MMs. Alternatives from commercial vendors include to receive MMs as XML data via a HTTP interface ▶[goyya.com :a02, p. 1] and from a MM7 interface ▶[Whatever Mobile :a02, p. 1].

How to send MMS from a web application. For sending MMs, there are several alternatives:

- **MM7.** A SOAP interface for the communication between MMS Relay/Server and MMS VAS Applications, standardized in ▶[3GPP :a01, pp. 127–157]. It is offered by mobile network operators, but accessing it directly seems reasonable only for major clients (VASPs). There are however commercial vendors who "loop it through" to end customers, e.g. the Whatever Mobile GmbH ▶[Whatever Mobile :a01, p. 1]. For those who want to access this interface, the Mbuni project ▶[mbuni.org :a01] is an open source solution to do so.

- **MM3 e-mail.** In terms of the MMS standard, a POP3 or IMAP4 interface to MMS is accessing MMS on reference point MM3 from an "external messaging server" ▶[3GPP :a01, p. 102]. The mapping between e-mails and MMS is standardized in RFC 4356 ▶[The Internet Society :a01]. There are several commercial vendors offering this interface (e.g. ▶[Whatever Mobile :a01, p. 1]), but of course mobile network operators are quite reluctant to do so for free.

- **MM3 HTTP.** Just like a SMTP interface, the MMS standard terms this a MM3 interface to an "external messaging server" ▶[3GPP :a01, p. 102]. The MM3 specification does not constrain the interface of external messaging servers, so the various proprietary HTTP interfaces offered from commercial vendors (e.g. ▶[goyya.com :a01]) are valid MM3 HTTP interfaces.

- **WAP push delivery.** This is no interface to MMS delivery protocols like MM3 or MM7 but an alternative mode of delivery directly to mobile phones, in circumvention of the mobile network operator's MMS. The basic

idea is to place a MM on a webserver and use a WAP push SMS to make a mobile phone retrieve this MM from its URL. This interesting idea is quite favorable to low-cost and pragmatic approaches to MM messaging. It is e.g. implemented in Stefan Hellkvist's little MMSLIB PHP library ▶[Hellkvist :a01]. This solution saves delivery costs with special GPRS rates, but without them the costs are far higher than compared to the delivery over a mobile network operator's MMS.

Regarding their fitness for being used in a prototype, the MM7 interface seems quite complex and WAP push delivery is quite expensive without GPRS flat or volume rates (which testers cannot be expected to have). The MM3 e-mail and HTTP interfaces are however quite adequate. CGW:RP eventually uses a simple proprietary MM3 HTTP interface, offered by a German vendor ▶[goyya.com :a01]. An order was placed to this vendor to acquire a contingent of MMs for testing CGW:RP. It is however comparatively easy to exchange the interface that CGW:RP uses for sending MMs: reimplement `MultimediaMessage::send()` to adapt to a new MM3 HTTP interface.

6 Conclusion: feedback, summary and outlook

The community stagnates without the impulse of the individual. The impulse dies away without the sympathy of the community.

attributed to William James

Summary. The MC³ is exposed to the judgment of several people, arriving at a preliminary valuation and some proposals for commercial exploitation. After that, this thesis concludes with a summary of its results, a presentation of its results in the context of current research and finally with an outlook on the next steps to further develop and apply this research work.

6.1 Discussion of the MC³

Summary. Some feedback on the proposed design of CGW is collected. Sources of feedback include personal experiences while developing the prototype CGW:RP and experimenting with it, reactions from a few non-developers who made experiences with the prototype and discussions about the MC³ with some others, including Ortwin Kartmann, manager of Promido Internet GmbH in Butzbach, Germany. The feedback is evaluated to arrive at a preliminary valuation of the MC³, some possibilities to improve it and hints on what will be important in UI design for CGW and its implementation.

MC³ from the user's perspective. An innovator should be able to clarify his or her idea and its benefit to potential customers and partner companies in no more than two or three sentences[50]. From several attempts to explain this thesis' idea to potential users it seems that a concise and intuitive explanation should use the metaphor of voice-based instant messaging by means of the mobile phone:

"CGW is a program which you can access with your mobile phone to maintain an ongoing conversation with a group of friends. It is like instant messaging with your voice, so the conversation will not bother you even if you keep it up all day long. In addition, the program has some features which can help you and your group to coordinate, organize and memorize."

[50]Thanks to Ortwin Kartmann for this valuable pragmatic piece of advice from business practice.

On the difficulties of agile collaboration. The ▶"Agile message" pattern is agile as it supports change-intensive coordination through reconfigurability. This is achieved by abandoning predefined static structures: agile messages and their various interactions are the only structural elements, and they are freely configurable. This approach can be compared to wiki systems, where wiki pages and their links are the main structural elements[51]. Ortwin Kartmann reported that a wiki system is used as the central knowledge management tool in his company, and that maintaining order in this system is really difficult. He traced this back to the lack of predefined structure in wiki systems, which means that structure has to be established based on conventions. Though limiting flexibility, features that introduce predefined structure have some definite advantages. He added a comparison between agile messages and the objects of object-oriented software development, pointing out the considerable amount of experience demanded by the latter. Based on these reflections, Ortwin Kartmann advised to provide a predefined example configuration within CGW, i.e. some agile messages which act as task pools, calendars etc. in a manner which makes sense for most users.

In addition, one should add that agile methodology is not "the correct approach for all things at all times […] [it] is a new option that needs to be understood and applied when the benefits are important." ▶[Dove :a02, p. 11_{+1}]. In other situations, especially for unaltered recurring tasks, solutions with more intrinsic structure could prove to be more efficient and less complex.

A related problem is that the MC³ is not that intuitively comprehensible: its arbitrary connections between artifacts can be understood with the graph metaphor, but this will be only at hand to the intuition of people with a mathematical background. Compared to that, a spatial metaphor is fit for intuitive apprehension by most people. This metaphor would employ proximity and containment relationships; for example, Xu's artifact-based CSCL system "Lecture 2000" uses a spatial approach ▶[Xu :a01, pp. 67-98_{+12}].

On conversation-style communication. When discussing the MC³, people were ready to admit that message-based communication offers a great

[51]Ward Cunningham invented "WikiWikiWeb", the first wiki ever ▶[Cunningham et al :a01] (cited from his first wiki site ever). The fact that he is at the same time a pioneer of agile software development ▶[Beck et al :a03] is interesting but should not serve as a premature indicator that all agile collaboration tools must contain only one type of connectible structuring elements.

advantage over synchronous communication technologies like phone calls or PTT. However, one participant thought it necessary for the group to know the point of time when messages are delivered to group members.

This necessity was confirmed by experiences with CGW:RP. In one instance during the tests, a MM arrived while the recipient just talked to somebody over his mobile phone. Because of this, the acoustic signal of the arriving MM was either missed or not played at all. Additionally, he missed the optical reminder on the mobile phone's default screen. For some time, the sender was unsure if the MM has arrived and eventually used a phone call to point out the MM to the recipient.

Reliable message-based communication seems impossible without per-default receipt notifications. These notifications would be a fixed feature of the groupware and could not be turned off individually as with e-mail today. The feature should become a part of the ▶"Conversation paradigm" pattern because in face-to-face communication it is likewise impossible to plausibly deny receiving an utterance. Generally speaking, in order to define a valuable and enjoyable mode of computer-mediated group communication it seems necessary to enforce some constraints wherewith the single individuals of the group will not be too comfortable.

Another problem indicated by CGW:RP's missed message notifications is that it will be difficult and frustrating to uphold a near-synchronous conversation if missing a message notification is not just the exceptional case. The mobile phone should exhibit a more intelligent and diligent behavior to ensure that a recipient reads and hears the message before its latest point of delivery.

On legacy technology and network effects. These issues form a complex problem which seems to be one of the biggest obstacles both for development and use of CGW. Before discussing it, here are the experiences with CGW:RP which raised this issue:

- **Phones with no MMS support at all.** Though the great majority of current new mobile phones is capable of MMS▶[52], there are phones in use which do not support it. Concretely, 2 of 5 potential test users for CGW:RP had a mobile phone without MMS support.

[52]In Germany, 83 percent of newly sold mobile phones were capable of MMS already in mid 2005 ▶[Heise Zeitschriften Verlag :a01].

- **Phones with limited MMS support.** Not all phones which claim to support MMS can communicate with CGW:RP. The reason is that some phones support not all content types allowed in MMs. In a particular case, CGW:RP sent a message to a recipient's Nokia handset, containing text and AMR audio. The message arrived as a MM which contained the text only; additionally, the recipient received a SMS notification which pointed to the mobile network operator's web portal in order to retrieve the remaining (audio) content. Phones with such limited MMS capabilities (supporting probably MMs with text and images only) are obviously useless when trying to interoperate with CGW:RP.

- **Phones with erroneous MMS support.** One case within the process of testing CGW:RP is of particular interest. The test user owned a Nokia 6230 handset with which she was capable to receive and open MM messages sent from other mobile phones. It received the MM from CGW:RP and successfully showed its subject line, but when trying to open the MM the mobile phone's firmware crashed. The phone would turn off and restart after some seconds automatically. This was reproducible, even with MMs which contained no audio part and only ASCII characters in their subject and body. As it was successfully tested to read and hear CGW:RP's messages with another handset without any problems it seems very probable that this Nokia 6230 suffered from a firmware bug. An extensive search in several large self-help Internet forums on mobile phone technology had no success in finding a description of this particular problem but indicated that early versions of phone firmwares often suffer from software faults. Updating the firmware of the Nokia 6230 is only possible at a Nokia Service Center ▶[Nokia UK Limited :a01]. Thus, erroneous MMS support is a significant obstacle to using a MMS based groupware.

These technical difficulties got in the way of testing group communication with CGW:RP. Such tests will need a considerable amount of time to program some workarounds into CGW:RP, update mobile phones' firmwares and to provide mobile phones to those members of a community who cannot receive or send MMs with voice messages yet.

Even from these limited experiences collected up to now it becomes clear how important it will be for CGW to interface with legacy technology. Relying on MMS

only is simply no alternative: CGW will not be delightfully integrated into community life if community members need to invest hours to study manuals, error messages and to update their mobile phone's firmware, or even need a new mobile phone. Confer community's emphasis on simplicity of technology (▶p. 34), integration of legacy technology (▶p. 36) and low-cost technology (▶p. 36). For a premium MMS service to succeed, only the individual customer needs a MMS capable mobile phone. But for a MMS based groupware application to succeed, all or nearly all members of a pre-existing group need MMS capable mobile phones.

This huge impact of network effects can be minimized by providing interfaces for legacy technology. In the ideal case, all community members with a mobile phone would have equal access to the groupware, and even those without a mobile phone could participate in a more restricted manner (e.g. using e-mail, a landline phone or a web interface). To provide equal access to all mobile phones, there should be a legacy interface which uses just phone calls and SMS. This seems the least common denominator of currently used mobile phones. To increase convenience and to lower the costs of communication, other interfaces to CGW would exploit additional capabilities of mobile phones where they are available: MMS, PTT, WAP and mobile Internet, DSR and others.

A draft of the proposed legacy interface: agile messages are represented by SMs and stored within the mobile phone; they contain a text part and a phone number to access the audio part. This phone number contains a message identifier as its substation part. Many mobile phones provide a number extraction feature so that it is very convenient to call numbers contained in SMs. To send a message, a user sends the text part as SM to the system and immediately receives a call from the system to record the audio part. To annotate agile messages with interactions, one would use a quick reference card to pick up a command and send then send it via SMS. Alternatively, one could use voice commands[53].

On designing audio-based user interfaces. One experience with CGW:RP was that at least some handsets play the audio part of MMs aloud by default. This makes sense when MMs are used e.g. as electronic greeting cards accompanied with image and music. But where the audio part contains content this could

[53] See the Sphinx project ▶[Carnegie Mellon University :a01] for a high-quality open source solution for speaker independent command recognition.

become obtrusive, annoying or embarrassing, depending on the people around. One should add to the ▶"Conversation paradigm" pattern: the sender has full control who will receive the original message, just as in fair face-to-face conversations.

As concerning audio quality, CGW:RP's audio quality was fully comparable to normal phone quality, according to one test person. The AMR files contained in MMs had acceptable size: a message of one minute takes just below 100 kByte even though CGW:RP uses the AMR codec with the highest quality (MR122). Judging from my personal experience, audio input seems far more convenient than writing SMS messages; however, I need to get accustomed to the thought that recipients will not just own my text messages but something as "personal" as words from my own voice.

A related issue of voice interfaces was raised by Ortwin Kartmann: many people feel reluctant to "speak with machines". They will probably not use voice commands in public places, at least where the mode of their voice makes it apparent that they talk to a computer. CGW can do without voice commands. But if people feel uneasy at recording messages (as some do when talking to a telephone answering machine), there could be a problem. CGW should probably be tuned to give the impression of a (latency-tolerant) conversation instead of voice mail. Ultimately, only an extensive field test will reveal if people will like this mode of communication or not.

A last idea for optimizing audio user interfaces, again from Ortwin Kartmann: it should be possible to search the audio parts of stored messages. No speech recognition is necessary here, just a fuzzy comparison of the audio snippet to search and the messages to be searched. Avoiding speech recognition in mobile phone communication is favorable as it avoids its inherent unreliability. This limited reliability is due to the limited and unpredictable audio quality offered by mobile phones ▶[Pearce :a01].

Preliminary valuation of the MC³. Though testing CGW:RP has been quite fragmentary and informal yet, it provided valuable feedback for further development; this may indicate that focusing on prototypes and feedback in the next phases of CGW development will pay off abundantly.

Concerning the "Conversation paradigm" pattern, the collected feedback did not suggest to break one of its constraints on communication. On the contrary, the

feedback encourages to resemble face-to-face conversation even more closely. Namely, it was suggested to add per-default receipt notifications (cf. ▶p. 113) and full addressee control (cf. ▶p. 116).

In general, it seems that the proposed communication features of CGW (the ▶"Conversational message" pattern and constituents) may meet with voluntary or even delightful adoption. The feedback contains no hints on the contrary. And there is the example of PTT, a new technology which allows to use the mobile phone as a two-way radio in push-to-talk operation. PTT enjoys popularity in some countries and is expected to spread in Europe as well ▶[Siemens AG :a01, pp. 2.7]. From the user's point of view, using a mobile phone according to the ▶"Conversational message" pattern is PTT combined with the unobtrusiveness of SMS.

The collected feedback however included criticism on the coordination-centric (or, more precisely, communication-automating) part of the MC³ patterns, i.e. on the ▶"Interacting artifact" pattern and its constituents. Not their existence met with criticism but their highly reconfigurable character (cf. ▶p. 112) and, as a related problem, their low intuitive comprehensibility (cf. ▶p. 112). While it was found that collaboration in communities is agile (cf. ▶chp. 4.1, pp. 53), further research seems necessary on how to create simple and intuitive concepts and interfaces for agile collaboration. If based on the ▶"Interacting artifact" pattern in its current shape, CGW would require technical and organizational skills that cannot be expected in community settings. Some important values of community life seem to have gone out of sight during the design process of this part of the MC³; these values include simplicity of technology (cf. ▶p. 34), aesthetics of technology and fun in using it (cf. ▶p. 35) and the low emphasis assigned to efficiency (cf. ▶p. 37).

So some further research work seems necessary to make the coordination-centric part of the MC³ fit for community use. In its present state however, the MC³ could afford a suitable model for agile CSCW in commercial settings. Agile organization is an innovative solution for many problems of the competitive enterprise ▶[Dove :a01, pp. 1-3], and the MC³ offers agile organization of human collaboration in a quite general shape.

6.2 Options for commercial utilization

Summary. Commercial options for using the community groupware concept to create a revenue are presented and evaluated. This section largely draws on discussing the issue with Ortwin Kartmann, manager of Promido Internet GmbH in Butzbach, Germany[54].

Options overview. Basically, there are the following options to make use of the community groupware concept from a commercial perspective:

1. Sell the concept.
2. Implement it and sell the software to companies.
3. Implement it and sell the software to end customers.
4. Implement it and run a portal.
5. Implement it and find a big partner company to cooperate with.
6. Develop it until the market is penetrated with necessary infrastructure.

Discussing the options. Selling the proposed community groupware concept is not feasible as it is publicly available in this thesis. Which means that all readers are equally welcome to implement it as it is or in a modified version. The remaining commercial options, which deal with implementing CGW, are therefore relevant for all of us alike.

Selling CGW to companies is an option only if these companies can use it to earn money themselves; which is determined by the remaining options. Selling the software to communities or their members has some limitations: it poses a considerable technical barrier before potential users because the installation process cannot be trivial; and it removes the unique selling point from operating a community groupware portal simultaneously.

Another option is to run a portal instead of releasing the software. This keeps the innovative features of CGW as the unique selling point and offers spontaneous participation to customers, without the technical annoyances of the installation process. However, marketing the service remains the key to success and the most difficult problem to solve: it needs a huge amount of time, money and personal connections.

[54] It is however not necessarily implied that this section reflects his opinions.

Therefore, a very valuable idea for marketing is to find a big partner company to cooperate with. For example, a mobile phone manufacturer or a mobile network operator could integrate a possibility to access CGW into the menu structure of mobile phones. The partner company could provide CGW cost-free to its customers as an added value, thus distinguishing itself from competitors. Contracting a partner company is so important that it can pay off even if the partner company gets a very significant share of the profits. However, finding a partner company which is willing and appropriate for this collaboration is exceedingly difficult without pre-existing connections. If all attempts to do so fail, one would be confined to run an independent portal and use other methods of marketing.

One method in marketing is to provide a basic service for free. It is unimportant for marketing to companies, but widespread and nearly indispensable when marketing communication services to private customers. Through the rise of the Internet, many communication tools and services are provided for free, so people are reluctant to use alternatives where a fee is required. One will need many customers who use the cost-free service to gain some who upgrade to the "premium" service. This reflects for example the policies of several major providers of free e-mail. The pricing for the CGW service will probably be comparable, too. The price would be a monthly fee; it would not have to include the costs of network traffic, just as e.g. e-mail services and Internet access are sold separately. However, it would have to include costs which originate from automatic actions of the groupware application, e.g. WAP push SMS in mobile communication. These costs have to be really low for the flat pricing model to be cost-effective; this is an important constraint to heed when choosing which basis technologies to use for the application (e.g., a MMS-based or Java-based application for mobile phones).

Marketing is not the only problem of introducing CGW, though. When introducing new communication technology, there are network effects implied. In this case, people need to be persuaded why to move from familiar technologies (like SMS and phone calls) to CGW. A community could not be persuaded to do so if it is too difficult to participate for one or more of its members. This will be the case as long as not all mobile network operators offer mobile data communication rates or technologies which can be utilized to send voice messages at low enough costs. For example, push-to-talk (PTT) is a promising technology but in

Germany there are yet only a few mobile phones and mobile networks which support it. Another example are mobile data flat rates: they are promising but too expensive to be used in average non-commercial settings yet.

When confined to market CGW by oneself it seems therefore fit to develop the application in combination with a small portal, waiting until the needed technology is in place and the market is really ready for adopting the product. Having accumulated some years of experience and development up to this point of time, this will give a significant advantage over competitors.

A first valuation of the options. To conclude: while every commercial project bears some risk, CGW bears extra risk due to network effects, marketing problems and the limitations of current technology. A way to handle them seems to invest moderately into the development and continuously into the marketing of a community groupware portal until the market is ready to take the product. And if CGW keeps its promise of delightful integration into community life it could turn out to be a self-marketing product. Marketing tries to make people keen on a product, but of course it is far better if the product does this all by itself.

6.3 Summary of the research

Summary. Summarizes the results of the conducted research work but does not contextualize them yet. The principal result is an innovative groupware design, adequate for use by communities. The individual results are presented along the chapters they appear in.

▶**1. Introduction.** An idea what to do is the first result because identifying a problem to solve is a problem itself. The idea of this thesis is that adequate CSCW technology can provide assistance and increase flexibility in communities, up to a degree that changes lifestyle. This idea was verbalized as a vision, a set of claims and a product idea.

▶**2. Background.** An overview of sociological methods to study and explain collaboration has been achieved. The two competing organizational paradigms for collaboration have been studied in domain independent manner: the linear and the agile paradigm. Some important aspects of current CSCW technology and

voluntary technology adoption have been highlighted. A major result is the detailed model of community life which informed the design of CGW. It covers nearly 40 aspects of community organization, preferences, work and communication and was collected mainly through an extensive literature study. Its general statement is that community life is quite different from the work setting, and that these differences must affect CSCW design.

▶**3. Method.** Following some reflections, this chapter concludes that an abstraction-guided engineering process is adequate for designing a community groupware in reasonable time. The proposed method takes its first step in finding the current organizational paradigm which underlies community life. The second step is to implement this abstract model to fit the constraints of community life, technical equipment and software design.

▶**4. Results.** The agile organizational paradigm was found to be a proper abstraction of community organization, while the linear organizational paradigm was not. Correspondingly, the MC³ was developed as a CSCW model for agile groupware. It is a complete and elaborate proposal of features, consisting of twelve interaction patterns of which all (with the exception of ▶"Agile message") are re-usable beyond community groupwares. ▶"Agile message" is the proposed central feature: it melts communication and coordination into one. It includes the automation support for some communicative acts as proposed in pattern ▶"Interacting artifact". In addition to that, it includes the conversation-style, latency-tolerant mode of voice messaging proposed in pattern ▶"Conversational message".

▶**5. Implementation.** An early prototype of CGW was developed. The prototype uses MMS as its basic technology, which means that the development process afforded some interesting insights on interfacing with MMS from a computer which is connected to the Internet.

▶**6. Conclusion.** Feedback on the MC³ was collected from discussing it and from experiences with the CGW prototype. This feedback indicated that MC³-based groupware can be quite successful as a communication support tool in community life, but the coordination-support functionality is too complex for the community context yet. A discussion of commercial options was appended, somewhat indicating that developing CGW with moderate effort seems both

necessary and promising. This chapter closed with this summary, a contextualization of the results and an outlook on further activity.

6.4 Future research

Summary. This thesis introduced the idea to build community groupware, a new class of CSCW applications. Background, concept and an early prototype have been contributed but many other research tasks remain open before CGW becomes real. This chapter details these tasks and marks out the most urgent.

Overview of open research tasks. ▶Object 1 (p. 5) spanned a research space, and this thesis covered 8 of its 28 items. Which means that some work has to be done before CGW is a mature software product. Some tasks seem to be especially valuable for supplementing this thesis' results and for the next steps of community groupware research. These are, starting with the most important:

1. **Extend CGW:RP with support for legacy phones.** Additionally, the patterns ▶"Conversation paradigm" and ▶"Negotiated synchrony" should make their way into the prototype.
2. **Perform extended user acceptance tests with the updated CGW:RP.** The minimum duration for this first well-founded valuation of the MC³ should be several weeks within an already existing community.
3. **Experiment with new UI concepts and usability.** Audio-centric interfaces are not that widespread so that some usability research work will have to be done. And there is plenty of room for experiments with new audio and multimodal UI concepts, e.g. the proposed audio search without speech recognition (cf. ▶p. 116) or command-based speech recognition, made reliable for mobile phone use by transmitting high-quality audio messages and not audio streams which might be affected by transmission errors ▶[Pearce :a01].

Further ideas. As a new kind of CSCW application, CGW will hopefully elicit many unanticipated usage patterns and applications. Ideas for this are not too hard to find. One that may deserve extra research work is the integration of mobile community groupware with community websites and stationary collaboration tools like community displays. An interesting community display

system is e.g. DYNAMO from the Universities of Sussex and Nottingham; see ▶[Brignull :a05], ▶[Brignull et al :a03], ▶[Brignull et al :a04].

6.5 Contribution of the research

Summary. This thesis contains both CSCW research and industrial design. Its contributions to both domains are summarized here. The contributions to CSCW research are the innovative aspects of the MC³'s patterns; the contribution to CSCW industrial design is a yet unseen combination of features, constituting adequate support for community collaboration. Concluding, the results are placed in a wider context, which includes some remarks on tailoring CSCW technology to the needs of *humans*, not machines or organizations.

Innovative contributions to CSCW concepts. The MC³ includes some well-known ideas, namely ▶"Connectable artifact", ▶"Message", ▶"Subscription", ▶"Unbound communicator" and ▶"Universal member"; these are simply placed into the contexts of agile collaboration and community and justified to be appropriate there. But each of the other patterns contains innovative aspects and brings them in to CSCW research:

- ▶**Activity set.** A key idea is to use artificial restrictions in the user interface to better collect information which can be exploited later for social navigation.
- ▶**Agile collection.** A non-hierarchical, intuitive and self-adapting alternative to hierarchical navigation of items.
- ▶**Agile message.** This pattern accumulates the innovations of all other patterns and adds one innovative aspect itself: it argues that coordination should remain an essentially human task in CSCW tools. While the significance of communication for coordination is generally accepted, it is rarely questioned if automating coordination can be reasonably achieved with current technology.
- ▶**Conversational message.** Proposes sophisticated audio instant messaging. While both audio telecommunication and text-based instant messaging are well-known techniques, the combination seems to have been disregarded in spite of its obvious advantages. Voicemate's TotalReach℠ application ▶[Voicemate Inc :a02] is one of the few audio

messaging applications that goes beyond the traditional usage of voice mail; however, it is a notification tool, no communication tool at all.

- **Conversation paradigm.** This pattern is remarkable as it was approved by the collected feedback (cf. p. 116). It is innovative (if not original) in that it proposes to actively shape the mode of computer-mediated communication by artificial constraints which resemble face-to-face conversation. Many of these constraints have been applied elsewhere, but their collection into a single interaction pattern is probably new. A particular contribution to CSCW research is the identification of a well-designed "communication architecture" as necessary for improved computer-mediated communication. There will be alternatives to the conversation paradigm, but in all cases new media should have coherent and designed qualities. On the contrary, to enable anything which is possible with a new medium in an unmindful way can give raise to social acceptance problems.
- **Interacting artifact.** The innovative aspect here is to base CSCW tools on the automation of communication, not of coordination, and to effect coordination through communication. Some of the proposed interactions which automate communicative acts seem to be innovative ideas; namely, "materialized reminder" (cf. p. 91) and "distribute as awareness information" (cf. p. 91).
- **Negotiated synchrony.** This innovation is a means to combine synchronous and asynchronous communication into one medium; the degree of required resp. accepted synchrony is negotiated on an ongoing basis and without much overhead. The need for this kind of communication is acknowledged (e.g. [Covi et al :a01, p. 339$_{-338}$]) but it seems that there are few ideas yet how to accomplish it.

Innovative contributions to CSCW industrial design. Though the MC^3 proposes some new CSCW concept as presented above, these do not amount the character of the proposed groupware. Appropriateness for community use is achieved primarily by a mindful combination of several existing ideas. The resulting groupware is innovative in that it exhibits advantages over other modes

of mobile communication which could be utilized for community collaboration also[55]:

- **Push to Talk over Cellular (PoC).** Features of PoC which are useful for community collaboration are the elaborate support for group communication, the audio-centric user interface and the high degree of fidelity to the conversation paradigm; cf. [Siemens AG :a01, pp. 3-5]. It is however designed for near-synchronous communication only and does not allow to negotiate synchrony: messages are delivered immediately [Siemens AG :a01, p. 4]. So people will disconnect from the PTT conversation where unobtrusiveness is required. This means that an ongoing conversation throughout the day cannot be provided with PTT. And of course, PTT does not include dedicated groupware functionality comparable to the MC³'s automation of communicative acts in support of group coordination (cf. pattern "Interacting artifact").
- **Mobile instant messaging.** Text-based instant messaging moves from the desktop to the mobile and gains popularity there. One example is Agilemobile.com's Agile Messenger. It even includes PTT-like functionality [Agilemobile.com :a01], i.e. people can send voice messages which are played instantly or when accessing them, depending on the recipient's configuration settings. It is by design an instant messenger, missing the groupware functionality of the MC³: elaborated group communication support, synchrony negotiation, some constraints of the conversation paradigm, automated communicative acts to support group coordination, and support for legacy equipment.
- **Instant messaging for group collaboration.** The Five Across Collaborate™ software is dedicated to support group collaboration by integrating instant messaging, presence control, file sharing and project management; regarding the instant messaging functionality, it offers nice features like group messaging, offline messages, forwarding messages to SMS, archiving chats and conducting online polls [Five Across Inc :a01]. It is however not designed for the mobile context; compared to that, the MC³ offers convenient input on mobile phones through voice messaging. Also,

[55]Judging from my current personal knowledge of alternative technologies, the MC³ seems to provide the most appropriate solution for community collaboration. It was quite interesting to receive some confirmation from alternative technologies that have some features in common with the MC³ and came to my knowledge after its development.

the MC³ adds synchrony negotiation, the conversation paradigm and communication automation to support agile coordination better.

- **Xpress Audio Messaging.** This is a mode of using MMS, introduced by Nokia. It is distinguished by a usability-optimized user interface to record, edit and send audio messages as MMs, and Nokia even makes it possible to send these audio messages to legacy phones not capable of MMS ▶[Nokia :a01]. This technology seems equivalent to an improved version of CGW:RP but like other messaging technologies misses the features of MC³ which are dedicated to assist group coordination.

Intended effects on CSCW research. First of all, this thesis wants to draw attention to community groupware as a neglected type of application. Communities are currently supported with various kinds of phone and messaging technology, blogging applications, communityware and community displays (cf. e.g. ▶[Brignull :a05]), but groupware applications have been specific to the work context up to now. This thesis is intended to fill this gap with a reasonable design proposal.

This research work is also intended to elicit a discussion about some of the unconventional solutions proposed here:

- Is it indeed necessary to leave technical possibilities unused in favor of a "communication architecture" like the conversation paradigm?
- How appropriate is it in community context if coordination remains a human task, supported by automated communicative acts? How appropriate is it in work context?
- What can be said from real-world experiences about integrating coordination seamlessly with communication?

And for the broader context, this thesis is concerned about possible unhealthy effects of computer-mediated communication: automating coordination might deprive of social contact. In community context this would simply be rejected, but in work context it is just as well undesirable to hand over people to an increasing slavery which is executed by machines. I personally believe that the ideal and enjoyable setting to work and live in is rich of interpersonal communication. Where and only where coordination automation increases time and possibility for interpersonal communication I welcome it.

Viewing communication-intensive (agile) collaboration not as a blockade of efficiency but on the contrary as an ingredient for higher quality work is a perception shared by proponents of agile software development methodologies and expressed in the history of their Agile Manifesto:

> But while the Manifesto provides some specific ideas, there is a deeper theme that drives many, but not all, to be sure, members of the alliance. At the close of the two-day meeting, Bob Martin joked that he was about to make a "mushy" statement. But while tinged with humor, few disagreed with Bobs sentiments that we all felt privileged to work with a group of people who held a set of compatible values, a set of values based on trust and respect for each other and promoting organizational models based on people, collaboration, and building the types of organizational communities in which we would want to work. At the core, I believe Agile Methodologists are really about "mushy" stuff - about delivering good products to customers by operating in an environment that does more than talk about "people as our most important asset" but actually "acts" as if people were the most important, and lose the word "asset". So in the final analysis, the meteoric rise of interest in and sometimes tremendous criticism of Agile Methodologies is about the mushy stuff of values and culture. [...] Jim Highsmith, for the Agile Alliance ▶[Highsmith :a02]

A Glossary of terms and abbreviations

Summary. This glossary defines a controlled vocabulary. In many other cases, a glossary is employed as a collection of hints to help the reader understand a new topic. This is a purpose here, too. The glossary entries refer the reader to relevant primary and secondary sources which support quick apprehension of the explained concepts. Additionally, however, this glossary restricts the meaning of words in this thesis.

3G

The third generation of technologies for terrestrial mobile telephony. In many instances used as a synonym of UMTS technology. From the end user's perspective, a faster data transfer rate is the main improvement of 3G.

Activity set

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"An activity set is a set of items which a user has opened simultaneously, and it is a subset of a navigable set. The items in an activity set are supposed to be mutually relevant, i.e. activity sets can be used in social navigation to recommend relevant navigational alternatives."

adaptability

Within the scope of this thesis, a synonym for agility.

affordance

Originally introduced by the perceptual psychologist J. J. Gibson, meaning an action possibility that exists latently in the world, in the form of a relationship between the environment and an actor ▶[Norman :a11]. An affordance exists even when the actor does not yet know it. Donald Norman used the term in the sense of "perceived affordance" and made it popular in industrial design; but he clarified later that the original meaning is what was stated above ▶[Norman :a11].

A Glossary of terms and abbreviations

Agile collection

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"An intuitive, apprehensible, low-maintenance means to manage a change-intensive item set and to navigate it semantically. The set of items and its subsets are flat lists; subsets can include other subsets, which enables automatic change propagation."

agile design principles

Ten design principles for agile systems, derived from observations in the manufacturing industries and shown in ▶[Dove :a04, p. 9.6$_{-9.2}$]. Confer also ▶object 7 (p. 26) for a full quotation. Rick Dove termed them "Rrs principles", meaning "Reusable – Reconfigurable – Scalable" ▶[Dove :a04, p. 9.6$_{-9.2}$], but also "design principles for highly adaptable business systems" ▶[Dove :a04, p. 9.3$_{-9.2}$], "principles for agile systems" ▶[Dove :a04, p. 9.4$_{-9.2}$].

Agile message

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"The central feature for community groupware, perhaps re-usable for other applications. It integrates communication and coordination seamlessly: coordination is effected through communication and supported by some automated communicative acts. This pattern respects the requirements of community life to a high degree. It results in an application that runs on mobile phones and relies on voice input for the most part."

agile organizational paradigm

An organizational paradigm known from the manufacturing industry and from software engineering. It employs agility for the task of organizational design. A central idea is to provide high adaptability to unanticipated changes.

agile paradigm

A shorthand synonym for "agile organizational paradigm".

agility

The quality of a system to be able to adapt to unanticipated changes quickly and successfully. A more elaborate definition is contained in this thesis (cf. ▶p. 24).

A Glossary of terms and abbreviations

AMR

The adaptive multi-rate speech codec used in mobile telephony. It is specified mainly in 3GPP TS 26.071 ▶[3GPP :a02] and accompanying technical specifications in the 3GPP TS 26 series. It supports eight bit rates, namely MR475 (lowest, equivalents to 4.75kbit/s), MR515, MR59, MR67, MR74, MR795, MR102, MR122.

analytic model

A model to depict an existing system. Used in most cases to understand and study this system or to communicate about it. Contrast synthetic model.

awareness

In the CSCW context, the perception of the group's situation, including e.g. the activities and location of other group members, or past events. Awareness is provided implicitly in co-located collaboration but must be provided in distributed collaboration by a CSCW tool.

behavior-setting

A concept introduced in Barker's 1968 work "Ecological psychology" ▶[Barker :a01]. A behavior-setting is an ecological unit, existing by itself and for a prolonged period of time at a particular place. It consists of a milieu and a standing pattern of behavior which is encompassed by it. Milieu and behavior of a behavior-setting have a synomorphic relationship.

CGW

Abbreviates "community groupware". "CGW" denotes the software product envisioned in this thesis, i.e. the groupware based on the MC³. CGW is a CSCW application for communities, offering a mobile interface and using agility as its underlying organizational paradigm. It is designed for being integrated with delight into community life. The term "community groupware" however denotes the class of software products of which CGW is just one instance.

CGW:RP

Abbreviates "community groupware research prototype". The prototype of CGW which is implemented within this thesis.

A Glossary of terms and abbreviations

collaboration

Collaboration and communication are intersecting subsets of human interaction. Collaboration focuses on physical objects or shared goals and is explicated in the physical space. Collaboration intersects with communication because participants need to coordinate their actions in order to collaborate, and this coordination can be effected through communication.

combining pattern

Within the context of this thesis, it denotes a pattern which consists mainly of a combination of other patterns.

communication

Communication and collaboration are intersecting subsets of human interaction. Communication focuses on the communicating partners and their relationship and is explicated in the symbolic space. Coordination can be effectuated through communication.

community

Elizabeth Mynatt *et al.*, working in the domain of network communities, offer the following as a loose consensus: "a multidimensional, cohesive social grouping that includes, in varying degrees: shared spatial relations, social conventions, a sense of membership and boundaries, and an ongoing rhythm of social interaction" ▶[Adler et al :a01, p. 311$_{-309}$]. In an even broader sense, it is an aggregate of individuals with shared characteristics. The shared element can refer to things as diverse as situations, interests, values, ancestry or profession, for example.

In this thesis, "community" is used mostly in a more restricted sense. It refers to intentional communities, i.e. groups of people who know that they are in a community and know their fellow members from communicating with them or meeting them. It refers not to pure network communities. And it refers only to associations where activity is not paid for, thus excluding companies and many non-profit organizations. "Community" as used in this thesis includes e.g. student groups, neighborhoods, flat-sharing communities, churches, groups of friends, sports and leisure groups and so on.

A Glossary of terms and abbreviations

community groupware

A class of software applications which are specially designed to be used as groupware within communities. One (perhaps the first) member of this class is CGW.

Pure communication tools adequate for community use (such as PTT) are not community groupware as they lack a CSCW tool's dedicated support for coordinating activity towards a shared goal.

Communityware is not community groupware either. It is adequate for the moderate pace, asynchronous collaboration in pure network communities. Community groupware however is intended to support communities with some degree of co-location; here, activities have higher interdependences as they can deal with shared physical objects rather than just information.

communityware

A class of social software that supports primarily social interactions within large groups of people, but not, as groupware does, collaboration towards shared goals. Xu puts it thus: "communityware focuses on an earlier stage of collaboration, i.e. group formation from a wide variety of people" ▶[Xu :a01, p. 3_{+12}].

Connectable artifact

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: "▶Software applications offer unintegrated partial models of reality while reality offers seamless integration. As a solution, it is proposed to handle model elements generically, using a common interface called "connectable artifact". This interface resembles the character of real-world things and their connections."

controlled vocabulary

In the standard ANSI/NISO Z39.19-2005, it is defined as follows:

> A list of terms that have been enumerated explicitly. This list is controlled by and is available from a controlled vocabulary registration authority. All terms in a controlled vocabulary *must* have an unambiguous, non-redundant definition.
> *NOTE: This is a design goal that may not be true in practice; it depends on how strict the controlled vocabulary registration authority is regarding registration of terms into a controlled vocabulary.*
>
> At a minimum, the following two rules *must* be enforced:

1. If the same term is commonly used to mean different concepts, then its name is explicitly qualified to resolve this ambiguity.
   NOTE: This rule does not apply to synonym rings.

2. If multiple terms are used to mean the same thing, one of the terms is identified as the preferred term in the controlled vocabulary and the other terms are listed as synonyms or aliases.

▶[NISO :a01, p. 5_{+12}]◀[56]

Conversational message

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"Immediate communication is very helpful for the initiator but obtrusive for the targeted individual. This can be balanced by introducing message-based conversation, which exhibits a near-synchronous character. When using voice input, message-based conversation is fast and convenient even with mobile devices."

Conversation paradigm

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"Text-based CSCW suffers from communication problems unknown in face-to-face conversation. As a solution, conversational style is resembled. Copresence is however not enforced as it can be too obtrusive at times and is unnecessary."

coordination

Malone and Crowston studied different definitions and arrived at the following: "Coordination is managing dependencies between activities. […] This definition is consistent with the simple intuition that, if there is no interdependence, there is nothing to coordinate. It is also consistent with a long history in organization theory of emphasizing the importance of interdependence"◀[57] ▶[Crowston et al :a01].

coordinating representation

Introduced by Richard Alterman *et al.* and characterized as follows:

> Within a community of actors, designs that organize (structure) behavior in recurrent situations of cooperation develop over time. […]. In non face-to-face interactions, structures that simplify the coordination of a conventional behavior are coded into a *coordination representation*. The

[56]emphasis and de-emphasis in original

[57]emphasis removed from original

coordinating representation helps the participants to jointly make sense of the situation in the absence of a face-to-face interaction. ▶[Alterman et al :a03, p.43$_{-42}$]

CSCL

Abbreviates “Computer Supported Collaborative Learning”.

CSCW

Abbreviates “Computer Supported Collaborative Work”. Computers can support the following aspects of group work: its performance (e.g. through automation), its distribution across space and time (e.g. through telecommunication mechanisms), and the group’s knowledge management.

CSCW model

An organizational model that is concrete enough to be directly implemented as the feature set of a CSCW implementation.

DSR

Abbreviates “distributed speech recognition”. An emerging standard for 3G mobile phones to enhance the reliability of speech recognition with mobile devices. With DSR, the feature extraction is performed on the mobile device and the actual speech recognition on a server. For a discussion, refer to ▶[Pearce :a01]. For the standards documents, refer to ▶[STQ AURORA :a01], ▶[STQ AURORA :a02].

environment paradigm

A better alternative than the tool paradigm to structure CSCW applications. With this paradigm, a CSCW application is not just an enabling tool for one or another activity, but provides the context of these activities also: namely, awareness of the current situation and the social context. For a more detailed discussion, cf. ▶p. 29 and ▶[Pankoke-Babatz :a01, pp. 15-16$_{+16}$].

ethnography

When applied to the context of CSCW, it is traditionally used as a synonym to ethnomethodology ▶[Crabtree :a12, p. 2], though the latter seems to be the more precise term. Ethnomethodology resp. ethnography is a sociological discipline that is not pre-occupied with methods, models, taxonomies and other means to interpret and account for human work and behavior. As Crabtree puts it: “[e]thnomethodology is not a method but a discrete analytic

sensibility" ▶[Crabtree :a12, p. 1]. It just describes the methods people use to interpret the world around them and asks how these methods contribute to the maintenance of social order.

In the CSCW area, so-called ethnographically informed design gained some popularity. It aims to better capture the requirements by providing an unbiased description of how collaboration is achieved in real-world activity. This description must however not serve as a direct list of requirements, but as a source of reflection when designing a CSCW system ▶[Pankoke-Babatz :a01, pp. 14-15$_{+16}$].

extreme collaboration

A term coined by Gloria Mark; it refers to "working within an electronic and social environment that maximizes communication and information flow" ▶[Mark :a01, p. 89$_{-88}$]. Further treatment is provided e.g. in ▶[Chachere et al :a01].

framework

A term from general systems engineering and one of the three ingredients for agile systems (cf. also unit and system):

> Framework: A set of standards constraining and enabling the interactions of compatible system modules. ▶[Dove :a04, p. 9.5$_{-9.2}$]

iCalendar

A standard for sharing calendaring information, standardized in RFC 2445 ▶[The Internet Society :a02].

industrial engineering

The application of organizational design to the structure of industrial companies.

inhabitants

In the behavior-setting theory ▶[Barker :a01], the people who currently inhabit a behavior-setting. They are no constituent part of a behavior setting, they just make it observable through their behavior which is determined by the behavior-setting. For a discussion of the behavior-setting theory in context of CSCW, see ▶[Pankoke-Babatz :a01, pp. 19-50$_{+16}$].

Interacting artifact

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: "How to implement artifact-specific features into a generic artifact-management application? This pattern proposes to tie behavior to the artifacts, remove artifact types and to allow unrestricted interactions between artifacts. This solution is then applied to artifact-based, agile groupware."

latest time of delivery

This term is used within the thesis with a special meaning. It was introduced in pattern ▶"Negotiated synchrony" (cf. ▶p. 97) and denotes in message-based communication the latest point of time a sender thinks is reasonable for her message to be delivered.

lean production

A system of production tuned to minimal resource usage and maximum output of goods. The goods are of high quality and offer some variety. Lean production is a Japanese invention, continually developed by Toyota since the 1950's ▶[Dove :a02, p. 10].

linear organizational paradigm

An organizational paradigm denoting the opposite of the agile paradigm. It is an organizational paradigm prevalent in today's groupware systems, but also in military organization, bureaucratic government organization and many enterprises. It is characterized by hierarchical structures, centralized instead of localized decisions, high vertical interdependence, few horizontal interactions, controlling processes with plans and predefined workflows and by organizational invariance over time. Confer also the discussion of its characteristics in ▶chp. 2.2.1 (pp. 20).

Instead of "linear", some use the term "disciplined" (e.g. ▶[Boehm et al :a01]). This however might provoke the misconception that the agile paradigm (as its opposite) is "undisciplined" in the sense of "anarchic". The term "linear" bears a connection to the "linear paradigm" as used in science, built upon the assumptions of order, reductionism, predictability and determinism. Thus, the term "linear organizational paradigm" directly addresses its lack of change management. And it indicates that this paradigm models projects in linear manner: using a full upfront analytic decomposition and not allowing

feedback to influence the plan. It also indicates that its opposite (the agile paradigm) models organization as a non-linear system, allowing unanticipated change. The connection of agility to non-linear and complex systems is further elaborated on in this thesis (cf. ▶p. 24).

linear paradigm

A shorthand synonym for “linear organizational paradigm”.

mandated adoption

A mode of technology adoption where the usage of technology is prescribed by some authority. Used in this way in ▶[Brignull :a05, p. 17_{+3}].

MC³

An acronym for “Model of Computer-supported Collaborating Community”. Spoken ['em 'si: 'kyu:bd] and written also as “MCCC”. It designates the main result of this thesis: a design model of a distributed community which collaborates in agile ways (cf. ▶chp. 4.2, pp. 60). The MC³ is presented as a collection of interaction patterns. And by the way: yes, the resemblance to mc^2, the famous equivalent of energy, is there by design.

Message

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶“Synchronous communication is obtrusive, especially if people are in reach anywhere, anytime. To solve this, make asynchronous message-based communication the default.”

milieu

In the behavior-setting theory ▶[Barker :a01], this denotes a concrete, real-world environment that belongs to a behavior-setting, as the “setting” where the behavior of the inhabitants takes place. The central finding of the behavior-setting theory is synomorphy between the milieu and the standing pattern of behavior of a behavior-setting, i.e. the standing pattern of behavior is independent of the inhabitants. For a discussion of the behavior-setting theory in context of CSCW, see ▶[Pankoke-Babatz :a01, pp. 19-50_{+16}].

MM

Abbreviates “Multimedia Message” which is a message sent over MMS. “MMs” abbreviates the plural: “Multimedia Messages”.

MM3

One of the eleven interfaces to the MMS, called “reference points” in the standard ▶[3GPP :a01]. MM3 is the interface between the MMS Relay/Server and external servers, e.g. e-mail messaging or SMS messaging:

> Reference point MM3 is used by the MMS Relay/Server to send Multimedia Messages to and retrieve MMs from servers of external (legacy) messaging systems that are connected to the service provider’s MMS Relay/Server. ▶[3GPP :a01, p. 27]

MM7

One of the eleven interfaces to the MMS, called “reference points” in the standard ▶[3GPP :a01]. MM7 is the interface between the MMS Relay/Server and commercial value-added-service (“premium”) applications:

> Reference point MM7 is used to transfer MMs from MMS Relay/Server to MMS VAS applications and to transfer MMs from MMS VAS applications to MMS Relay/Server. This reference point shall be based on SOAP 1.1 […] and SOAP messages with attachments […] using an HTTP transport layer. ▶[3GPP :a01, p. 28]

MMS

Abbreviates “Multimedia Messaging Service”. Designed as the successor of famous SMS, it offers sending text with unlimited length, images, sound clips, video clips and synthetic music. A MM is divided into “slides”; a slide might combine e.g. a text message with an image and accompanying music. The current standard defining MMS is 3GPP TS 23.140 V6.13.0 ▶[3GPP :a01].

model

A representation of a system which is not the system itself; the system’s units and interactions are represented by symbols. Models might be used as prototypes or surrogates of the system, to communicate about the system or for other purposes.

Models have a pragmatic aspect: a model covers just that part of a system which is relevant in the current context and represents it in a way which is useful in the current context. Thus, it is a simplification (an “intentional misrepresentation”) of reality.

There is a difference between analytic models (which try to understand reality) and synthetic models (which add to reality, like computer programs). This difference is fluent however as every synthetic model is built on an

implicit understanding of how reality works, otherwise it could not interact with reality successfully. This implicit understanding is an analytic model.

MSISDN

Abbreviates “Mobile Subscriber Integrated Services Digital Network”. It designates the phone number of a mobile phone and is effectively a phone number in international format but without leading “00”.

MUD

Abbreviates “Multi User Dungeon”, also “Multi User Dimension”. A consistent virtual world provided to multiple users over a computer network. MUD applications are for example multi-user role-playing games or educational environments. ▶[Adler et al :a01, p. 210–209] comments on the role of MUDs as community-supporting software.

multimodal

A user interface which offers different modalities for human-computer interaction. These modalities mostly use different senses, e.g. a visual modality like screen output would be combined with a voice modality like speech input.

Negotiated synchrony

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶“Continuous mobile instant messaging can be obtrusive in spite of its message-based character. This pattern provides a synchronism negotiation protocol to cope with this: recipients may accept only a certain level of synchronism resp. obtrusiveness; senders may override this for urgent messages.”

network effect

Also called, in less precise manner, network externality. It is a concept from economics and defined as follows:

> Network externality has been defined as a change in the benefit, or surplus, that an agent derives from a good when the number of other agents consuming the same kind of good changes. As fax machines increase in popularity, for example, your fax machine becomes increasingly valuable since you will have greater use for it. This allows, in principle, the value received by consumers to be separated into two distinct parts. One component, which in our writings we have labeled the autarky value, is the value generated by the product even if there are no

> other users. The second component, which we have called synchronization value, is the additional value derived from being able to interact with other users of the product, and it is this latter value that is the essence of network effects. ▶[Liebowitz et al :a01, p. 1]

network community

A globally distributed community, the direct opposite of a co-located community. Or, as As Phil Agre puts it:

> 'Community network' suggests 'given a (geographically localized) community, build a network', and 'network community' suggests 'given a (globally distributed) network, build a community'. [...] In each case the word 'community' points to an ideal of overcoming social distance, and in each case the word 'network' suggests an instrument for doing so. ▶[Agre :a01]

obtrusiveness acceptance level

This term is used within the thesis with a special meaning. It was introduced by the pattern ▶"Negotiated synchrony" on ▶p. 97. Not unlike the concept of "presence information", it provides awareness information on a person's ability or willingness to communicate in a certain way.

organizational design

The activity of developing an organization's organizational model. Within this process, design principles (like the agile design principles) and discoveries, patterns and experiences can be utilized.

organizational model

A model of an organization's organizational system. It might be an abstract organizational model (like organizational paradigms) or a concrete organizational model (like CSCW models). In the latter case, it consists of entities like hierarchies, departments, roles, access rights and so on. Concrete organizational models have an underlying abstract organizational model (an organizational paradigm), even if designed without awareness of it.

organizational paradigm

A structuring conceptualization of activity, sometimes described as "abstract organizational model" in this thesis. It determines the way we see or want to see the structure and connections of people, tasks, rights and information. The term refers to the basic and quite abstract principles of organization, as opposed to concrete organizational models like CSCW models. The following

instances are discussed in this thesis: the linear organizational paradigm and the agile organizational paradigm.

pattern

Patterns as a means to communicate expertise have been brought to the software development community by the Gang of Four book “Design Patterns: Elements of Reusable Object-Oriented Software” ▶[Gamma et al :a01]. John Vlissides gives an inclusive definition of “pattern” ▶[Vlissides :a02] which equivalents to: a pattern is a named solution to a recurring problem in a context and contains some teaching how to adapt the pattern to concrete situations. For a problem to be recurring it must be verbalized with at least a basic level of abstraction.

phatic communication

The use of socializing communication in order to build rapport. Usually, one uses set phrases which already lost their original meaning at the beginning and the end of a conversation. This is an example of phatic communication.

PLML

Abbreviates “Pattern Language Markup Language”. A proposal for a pattern format for human-computer interaction patterns, developed during a CHI 2003 workshop, introduced in ▶[Fincher :a02].

PoC

Abbreviates “Push to Talk over Cellular”, which see.

pragmatic action

▶[Alterman et al :a10] claims that human problem solving behavior is pragmatic rather than analytic. Alterman *et al*. then defines pragmatic action as follows:

1. The everyday task environment consists of designed artifacts, containing designed hints on problem solving. Pragmatic action means then to solve problems using the help of others, partially as coded into the task environment by them. This help is accessed in a pragmatic way: only when it is necessary, and only what is necessary of it.
2. Most task environments are semi-permanent. Pragmatic action means to organize behavior around these task environments and the

individual's experiences with them, accumulating these experiences to form skill.

See ▶[Alterman et al :a10, pp. 77–79$_{-52}$] for the triggered role of memory in everyday activity, rendering it impractical to develop full "project" plans for this sort of activity.

PTT

Abbreviates "Push to Talk", which see.

Push to Talk

Denotes a mode of operating voice telecommunication equipment: the equipment is in reception mode by default and a button is used to switch to transmission mode. PTT referred originally to two-way radio but is used currently also for a similar mode of operating mobile phones. The more precise term for this is "Push to Talk over Cellular", which see.

Push to Talk over Cellular

Operating mobile phones in PTT mode. In general, the user needs a phone with PoC support. Characteristics of PoC include:

- It is intended for group communication ▶[Siemens AG :a01, p. 3].
- To receive transmission rights, the user pushes and holds a button; as only one participant can transmit at a time, this does not necessarily mean that transmit rights are granted ▶[Siemens AG :a01, p. 4].
- Messages are transmitted packet-switched over mobile networks, so PoC does not suffer from limited transmission ranges as traditional two-way radio handsets ▶[Siemens AG :a01, p. 5].
- As transmissions are unidirectional only, PoC affords a much simpler implementation than VoIP telephony. Before sending a message, up to two seconds are employed for buffering. That way, there is no need for quality of service parameters in the packet-switched network ▶[Siemens AG :a01, p. 14].

real-time

An interaction within a dynamic system is "in real-time" if it is completed within the maximum delay allowed. For example: the pattern ▶"Negotiated synchrony" proposed in this thesis (cf. ▶p. 96) implements real-time

interactions, the “maximum delay” is specified via a “latest time of delivery”. Real-time operation is not necessarily fast: the “latest time of delivery” could be five hours away.

reified interaction

An interaction between system units of one system that appears at the same time as a system unit of another system. Reification is the view of treating something as a thing though it has not the character of a thing by itself.

response ability

Rick Dove, once Director Strategic Analysis of the Agility Forum, sees this as the key aspect of agile systems in the manufacturing industries ▶[Dove :a03, p. 2] and defines it thus:

> Being agile doesn’t mean being in control. It means having a controlled response ability to deal effectively with things that are beyond control—whether internal or external, whether opportunity or necessity. Response ability is obtained through culture and structure. [...] Response ability alone does not make an organization agile. It is necessary, but it is only a capability. In order for that capability to be employed effectively there are two more necessary elements: timely knowledge management and decisive value propositioning. ▶[Dove :a01, p. 2-3]

semantic navigation

Navigating between items based on their semantic relationship. These relationships might be mapped to a spatial layout, but the mode of navigation is nonetheless semantic. Semantic navigation occurs e.g. in hypertext systems where links are navigated based on their semantics and not based on their spatial position in the text. For this concept, cf. ▶[Chalmers et al :a02, p. 1].

sequential multimodal input

The simplest type of a multimodal user interface. At each point of time, only one input modality is allowed, depending on the state of the interaction. But different modalities may appear in sequence. The term was borrowed from ▶[Kvale et al:a01, p. 105$_{-103}$].

simplicity

The beautiful definition from “Principles behind the Agile Manifesto” says it is “the art of maximizing the amount of work not done” ▶[Beck et al :a04].

SM

Abbreviates “Short Message”, which is a message sent over SMS. Though not that far-spread, it is the abbreviation used in the SMS specification ▶[3GPP :a03]. “SMs” abbreviates the plural: “Short Messages”.

SMS

Abbreviates “Short Message Service”. A telecommunication service to transmit short text messages between mobile phones. It is standardized in ▶[3GPP :a03].

SMTP

Abbreviates “Simple Mail Transfer Protocol”. A wide-spread protocol for submitting and transferring e-mails.

SOAP

A protocol to transfer data and documents in distributed systems, and to execute remote procedure calls.

social navigation

A concept introduced by Dourish and Chalmers in 1994 as an alternative to semantic navigation:

> In social navigation, movement from one item to another is provoked as an artefact of the activity of another or a group of others. So, moving “towards” a cluster of other people, or selecting objects because others have been examining them would both be examples of social navigation. ▶[Chalmers et al :a02, p. 1]

sources of synomorphy

In the behavior-setting theory ▶[Barker :a01], this denotes mechanisms how synomorphy is achieved between the milieu and the inhabitants’ behavior of a behavior-setting. These sources include for example physical and social forces. For a more detailed (but concise) presentation, cf. ▶[Pankoke-Babatz :a01, pp. 34–41_{+16}].

standing pattern of behavior

In the behavior-setting theory ▶[Barker :a01], this denotes an invariant cohesive set of behaviors of inhabitants which is enforced by an behavior-setting through the sources of synomorphy between milieu and behavior. Confer ▶[Pankoke-Babatz :a01, pp. 34–41_{+16}].

Subscription

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"To cope with information overload and privacy concerns while maintaining openness of communication, provide the possibility to subscribe to groups (or rooms, channels, lists, …) and to dynamically create new ones."

synthetic model

A model to create an artificial system which is then added to the "world" system. Contrast analytic model.

SyncML

A language for data synchronization, invented for the synchronization of mobile devices (such as their address book entries, messages etc.). It is however applicable to every kind of data. An interesting open source project concerned with SyncML is Funambol ▶[Funambol Inc :a01].

sync point

This term is used within the thesis with a special meaning. It was introduced by the pattern ▶"Negotiated synchrony" on ▶p. 97. It denotes a point of time when a participant will have caught up with the messages in her message queue at latest. Participants provide this information as awareness information whenever they cannot receive all messages immediately.

synomorph

In the behavior-setting theory ▶[Barker :a01], this denotes a structure equivalence between milieu and one standing pattern of behavior.

system

A term from general systems engineering and one of the three ingredients for agile systems (cf. also framework and unit). The definition used in this thesis fits general systems engineering and comes from research about agile design principles:

> System: A group of interacting modules sharing a common framework and serving a common purpose. ▶[Dove :a04, p. 9.5$_{-9.2}$]

The generic character of this definition is emphasized:

> Any organization of interacting units is a “system”: an enterprise of business resources, a team of people, a cell of workstations, a contract of clauses, or a network of suppliers. ▶[Dove :a04, p. 9.6$_{-9.2}$]

task environment

An environment combined with a task (i.e. something requiring steps, like an aim or problem). The task allows to delimit the environment, sorting irrelevant things out. Introduced into cognitive science in 1972 by ▶[Newell et al :a01].

third place

A term coined by sociologist Ray Oldenburg in his book “The Great Good Place: Cafes, Coffee Shops, Bookstores, Bars, Hair Salons, and Other Hangouts at the Heart of a Community” ▶[Oldenburg :a01]. With work and the home being the first two places, the third place is where people can come to regularly and commune in company they like in relaxed manner.

tool paradigm

A possibility to structure CSCW applications which is prevalent in current applications. Within this paradigm, a CSCW application is a tool that enables to perform some kind of collaborative or coordinative activity, but neither are different tools integrated with each other nor do these tools explicate the group perspective on collaboration (rather than the individuals’) anywhere. For a more detailed discussion, cf. p. 29 and ▶[Pankoke-Babatz :a01, pp. 15-16$_{+16}$].

typeless artifact

This term is used within the thesis with a special meaning. It was introduced by the pattern ▶“Interacting artifact” on ▶p. 89 and denotes artifacts which do not differ in the structure of contained data but only in behavior. The concept of typeless artifacts is a candidate for a basic structure of generic artifact-managing applications.

UI

Abbreviates “User Interface”.

Unbound communicator

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"A problem of current telecommunication technology is failing connections because people are not where their equipment is. To co-locate equipment and communicator, make the equipment mobile and wearable."

unit

A term from general systems engineering and one of the three ingredients for agile systems (cf. also framework and system). A unit is a component of a system which has a defined and independent identity, capability and purpose and which is able to interact with other units. The original term used by Rick Dove in agile systems engineering was "module" (equivalently defined in ▶[Dove :a04, p. 9.5$_{-9.2}$]), but this was abandoned in the transition to more intuitive vocabulary; cf. the new verbalization of the agile design principles in ▶[Dove :a04, p. 9.6$_{-9.2}$].

Universal member

One of the twelve interaction patterns of the MC³ developed in this thesis. Its summary reads: ▶"Role-based organization malfunctions where people are not accessible permanently or the inflexibility of being accessible is unbearable. The alternative is to go towards role-free organization."

VAS

Abbreviates "Value Added Service". These are so-called premium services which third-party companies can offer in mobile networks. A well-known example of a VAS is the possibility to send a postcard via premium MMS.

VASP

Abbreviates "Value Added Service Provider", a provider of VAS.

vCard

A file format to store and exchange address book entries. Version 3.0 got standardized in RFCs 2425 ▶[Dawson et al :a01] and 2426 ▶[Dawson et al :a02].

VoIP

Abbreviates “Voice over Internet Protocol”. A technology which enables people to have a voice conversation which is mediated by an IP-based data network. Also called “IP telephony”.

voluntary adoption

A mode of technology adoption where users are allowed to choose for themselves if they want to use a new technology. Used that way in ▶[Brignull :a05, p. 17_{+3}].

WAP

Abbreviates “Wireless Application Protocol”. It is a set of protocols which allows to access parts of the Internet from mobile devices with limited capabilities, especially from mobile phones.

XC

Abbreviates “Extreme Collaboration”. This abbreviation is for example used in ▶[Chachere et al :a01].

XP

Abbreviates “Extreme Programming”. An agile software development methodology. On the phenomenological side, XP is centered around rapid feedback, testing, simplicity, small-scale changes and customer collaboration. For a further discussion, cf. the paragraphs on XP in this thesis (▶p. 27) and cf. ▶[Beck :a02], where XP is explained in great detail by one of its innovators.

B Source code of CGW:RP

Summary. The source code of the CGW research prototype developed within this thesis is shown here in full length. Headings have been inserted to aid navigation; to be executable, the source code must be placed into one file in the order it appears here. CGW:RP starts with configuration settings, then defines class MultimediaMessage as a reusable module to create and send MMs, adds some utility functions to handle e-mails, HTTP and FTP connections and AMR audio and concludes with the main program which receives and distributes MMs.

B.1 Instructions and configurations

<?php

```
/** CGW:RP - community groupware research prototype
  *
  * A small prototype of community groupware, implementing voice-based
  * instant messaging for mobile phones via MMS. Mobile phone users have
  * to address their MMs to a (configurable) e-mail address. This script
  * uses the MR122 AMR codec (12,2kBit/s) which means that a MMS of
  * 100 kByte might contain an audio message up to one minute. 100 kByte
  * is the maximum with the MMS service of massenversand.de at the
  * moment.
  *
  * Installation of this prototype:
  * Development and tests have been done with Linux, but as all used
  * third-party tools are available for Windows too (PHP, Apache, SoX,
  * 3GPP AMR encoder TS 26.104) it should be possible to port it with
  * ease. How to install at Linux / Unix:
  * (1) provide the command line PHP4 interface or a web server (tested
  *   with Apache 1.3.34) with PHP4; CGW:RP is never invoked from the
  *   Internet via an URL, so it is enough to install it on a local
  *   machine which has permanent Internet access but no URL attached
  * (2) provide the PHP4 extensions imap.so and curl.so
  * (3) compile and install the AMR encoder of 3GPP TS 26.104 V 5.2.0 and
  *   adapt the path to it; detailed instructions below
  * (4) install the audio converter sox (Sound eXchange)
  * (4) provide a CRON job which calls this script once a minute
  * (5) make sure the mobile phones of all participants are configured
  *   for MMS
  * (6) create an IMAP account and configure this script to access it
  * (7) create an FTP account and configure this script to access it
  * (8) get an account to send MMs from massenversand.de (or from a
  *   company with equivalent HTTP interface to send MMs) and configure
  *   this script to access it
  *
  * Compiling the AMR encoder:
  * (1) download 3GPP TS 26.104 from
  *   ftp://ftp.3gpp.org/Specs/latest/Rel-5/26_series/26104-520.zip
  * (2) unzip 26104-520.zip -d 26104-520; cd 26104-520;
  *   unzip 26104-520_ANSI_C_source_code.zip;
```

```
  * (3) cp makefile.gcc Makefile;
  * (4) open Makefile and remove the option -DETSI; this makes the
  *   encoder create AMR files in "AMR mime file storage format"
  * (5) open decoder.c and add: #include "sp_dec.h"
  * (6) make;
  * (7) install equivalently to:
  *   cp /var/www/cgw/26104-520/decoder /usr/local/bin/amr-decoder;
  *   cp /var/www/cgw/26104-520/encoder /usr/local/bin/amr-encoder;
  *
  * @author Matthias Ansorg \< matthias@ansorgs.de \>
  * @copyright Matthias Ansorg
  * @license public domain (but the author is not liable for anything
  *   whatsoever)
  * @file
  * @todo limit text length to 300 characters for each slide (else the
  *   message might not be accepted by mobile network operators)
  *   however, a text of 391 chars was transmitted without problems
  * @todo when necessary, adapt AMR codec and shorten AMR file to keep
  *   below the 100 kByte MM size limit; presently, users who send MMS
  *   messages must heed this constraint
  */

// adapt these constants and variables to configure the script
// include the path of commands if they reside outside of PHP's PATH
define('LOGFILE_NAME', __FILE__ . '.log.txt');
define('SOX_CALL', 'sox');
define('AMRENCODER_CALL', '/usr/local/bin/amr-encoder');
define('IMAP_SERVER', 'mail.example.org');
define('IMAP_PORT', '143');
define('IMAP_LOGIN', 'mms+example.org');
define('IMAP_PASS', 'example_pass');
define('UPLOAD_BASE_URL', 'http://mms.example.org');
define('UPLOAD_FTP_SERVER', 'ftp.example.org');
define('UPLOAD_FTP_USER', 'mms@example.org');
define('UPLOAD_FTP_PASS', 'example_pass');
define('MMS_USER', 'example_user'); // user name to send MMs
define('MMS_PASS', 'example_pass'); // password to send MMs
$participants = array( // all group members in format MSISDN => name
  '4912312345671' => 'Firstname1 Lastname1',
  '4912312345672' => 'Firstname2 Lastname2',
  '4912312345673' => 'Firstname3 Lastname3'
);
```

B.2 Class MultimediaMessage

```
class MultimediaMessage {
  var $mmSubject;
  var $mmTo;  // mobile phone numbers, international format ('00'.MSISDN)
  var $mmImgWidth = NULL; // if unspecified, the server assumes 160 px
  var $mmImgHeight = NULL; // if unspecified, the server assumes 120 px
  var $mmsUser;
  var $mmsPass;
  var $mmsServer = 'http://www.massenversand.de';
  var $mmsUrlPath = '/mmsexec/sendmms.asp';
  var $mmsLastResponses;
  var $mmsLastResponseCodes;
  var $mmsLastResponseDescs;
  var $mmSlides = array();
```

```
var $mmSlidesNextIdx = 0;

/** create a MM
  * @param $mmTo recipient mobile phone number in MSISDN format;
  *   or array of such numbers
  * @param $mmImgWidth all images of the MM are scaled to this width
  * @param $mmImgHeight all images of the MM are scaled to this height
  */
function MultimediaMessage($mmSubject, $mmTo, $mmImgWidth = NULL,
  $mmImgHeight = NULL, $mmsUser = NULL, $mmsPass = NULL)
{
  $this->mmSubject = $mmSubject;
  if (!is_array($mmTo)) $this->mmTo = array($mmTo);
  else $this->mmTo = $mmTo;
  if (isset($mmImgWidth)) $this->mmImgWidth = $mmImgWidth;
  if (isset($mmImgHeight)) $this->mmImgHeight = $mmImgHeight;

  // convert $this->mmsTo from MSISDN format to international format
  foreach ($this->mmTo as $toNo => $toName) {
    $mmToIntl["00$toNo"] = $toName;
  }
  $this->mmTo = $mmToIntl;

  if (isset($mmsUser)) $this->mmsUser = $mmsUser;
  elseif (defined('MMS_USER')) $this->mmsUser = MMS_USER;

  if (isset($mmsPass)) $this->mmsPass = $mmsPass;
  elseif (defined('MMS_PASS')) $this->mmsPass = MMS_PASS;
}

/** adds a slide to the MM
  *
  * Slides might contain image, text, sound and a duration. Images and
  * sounds can given as either absolute or complete URLs.
  * @todo scale the image to given width and height; this is necessary
  *   for a general MM handling class but not for its first usage here
  */
function addSlide($img, $txt, $snd, $duration = NULL) {
  // convert relative to complete URLs where necessary
  foreach (array('img' => $img, 'snd' => $snd) as $kind => $fileId) {
    if (isset($$kind)) {
      $fileIdParsed = parse_url($fileId);
      if (empty($fileIdParsed['scheme'])) { // not yet a complete URL
        if (substr($fileId, 0, 1) == '/') {
        // $fileId contains a full path relative to document root
          $$kind = "http://{$_SERVER['SERVER_NAME']}$fileId";
        }
        else { // relative path
          $$kind =
            "http://{$_SERVER['SERVER_NAME']}" .
            dirname($_SERVER['PHP_SELF']) .
            "/$fileId";
        }
      }
    }
  }

  if (isset($img))
```

```
      $this->mmSlides['image']    [$this->mmSlidesNextIdx] = $img;
    if (isset($txt))
      $this->mmSlides['text']     [$this->mmSlidesNextIdx] = $txt;
    if (isset($snd))
      $this->mmSlides['sound']    [$this->mmSlidesNextIdx] = $snd;
    if (isset($duration))
      $this->mmSlides['duration'][$this->mmSlidesNextIdx] = $duration;
    $this->mmSlidesNextIdx ++;
  }

  function send() {
    $this->mmsLastResponses = array();
    $this->mmsLastResponseCodes = array();
    $this->mmsLastResponseDescs = array();
    foreach ($this->mmTo as $mmToNumber => $mmToName) {
      $queryArray = array(
        'user' => $this->mmsUser,
        'pass' => $this->mmsPass,
        'subject' => $this->mmSubject,
        'to' => $mmToNumber
      );
      if (isset($this->mmImgHeight))
        $queryArray['height'] = $this->mmImgHeight;
      if (isset($this->mmImgWidth))
        $queryArray['width'] = $this->mmImgWidth;
      $queryArray = $queryArray + $this->mmSlides;

      $this->mmsLastResponses["$mmToNumber"] =
        http_post($this->mmsServer, $this->mmsUrlPath, $queryArray);

      $this->mmsLastResponseCodes["$mmToNumber"] =
        substr($this->mmsLastResponses["$mmToNumber"], 0, 3);
      $this->mmsLastResponseDescs["$mmToNumber"] =
        substr($this->mmsLastResponses["$mmToNumber"], 4, -1);
    }
    // return true if mmsLastResponseCodes contains only code 200 (OK)
    return(array_values(array_unique($this->mmsLastResponseCodes)) ==
      array('200'));
  }
}
```

B.3 Utility functions

```
/** http_build_query reimplementation for PHP versions before 5.0
  *
  * Adapted from a PHP user note at
  * http://www.php.net/manual/en/function.http-build-query.php
  */
if(!function_exists('http_build_query')) {
  function http_build_query(
    $formdata, $numeric_prefix = NULL, $key = NULL
  ) {
    $res = array();
    foreach ((array)$formdata as $k=>$v) {
      $tmp_key = urlencode(is_int($k) ? $numeric_prefix.$k : $k);
      if ($key) {
        $tmp_key = $key.'['.$tmp_key.']';
      }
```

```
      if (is_array($v) or is_object($v)) {
        $res[] = http_build_query($v, NULL, $tmp_key);
      }
      else {
        $res[] = $tmp_key.'='.urlencode($v);
      }
    }
    return(implode('&', $res));
  }
}

/** perform a HTTP POST request using the cURL PHP extension
  * @param $server where to POST to, e.g. http://www.example.org
  * @param $path URL part after server name, e.g. '/foo/bar.php'
  * @param $vars array of key/value pairs, maybe nested; or an object
  * @return the content returned by the server, without headers
  */
function http_post($server, $path, $vars) {
  $ch = curl_init();
  curl_setopt($ch, CURLOPT_URL, $server.$path);
  curl_setopt($ch, CURLOPT_POST, 1);
  curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($vars));
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
  $reply = curl_exec($ch);
  if (curl_errno($ch))
    error_log(
      "ERROR: curl_exec() error no ". curl_errno($ch) . " : " .
        curl_error($ch)."\n",
      3, LOGFILE_NAME
    );
  curl_close($ch);
  return($reply);
}

/** decodes a e-mail part to 8 bit text according to its encoding
  *
  * See http://www.php.net/manual/en/function.imap-fetchstructure.php
  * for the meaning of the encoding type numbers.
  */
function mime_decode($text, $encoding) {
  if ($encoding == 4) {
    $text = quoted_printable_decode($text);
  }
  elseif ($encoding == 3) {
    $text = base64_decode($text);
  }
  return($text);
}

/** enforces constraints on the subject of a MM
  *
  * If a MM subject does not comply with these constraints, sending
  * might be possible (at least with massenversand.de it is) but the
  * mobile network operator might not accept the MM (with
  * massenversand.de, "200 OK" without the (usual) size indication is
  * returned and the MM is not delivered to the recipient's mobile
  * phone).
  */
```

```
function mm_subject_constraints($subject) {
  // max. 36 characters allowed for the interface of massenversand.de,
  // max. 40 one-byte characters allowed per OMA specification
  $subject = substr($subject, 0, 35);

  // the characters ()[] definitely do not work, the characters
  // :-_.[:alnum:][:space:] definitely work
  $subject = ereg_replace(
    '[^-_\.\!@#&\*\+\=\|\?\:[:alnum:][:space:]]', '?', $subject
  );

  return($subject);
}

/** extracts the first part with given MIME type from a e-mail
  * @param $reqType an integer designating the type of the message, or
  *   use one of the constants defined in the imap extension: TYPETEXT
  *   (0), TYPEMULTIPART (1), TYPEMESSAGE (2), TYPEAPPLICATION (3),
  *   TYPEAUDIO (4), TYPEIMAGE (5), TYPEVIDEO (6), TYPEOTHER (7). But
  *   avoid TYPEMULTIPART (1) as it makes no sense to fetch such a part.
  * @param $subPartId extract from this part of the e-mail and its
  *   subparts; for the identifier format, see RFC2060, section 6.4.5.
  *   In short: '1' is the whole body, or for MULTIPART e-mails, the
  *   first part, 1.1 the first subpart of a first MULTIPART part in a
  *   MULTIPART e-mail etc.
  */
function extract_typed_part(
  $mailbox, $msgNo, $reqType, $thisPartStructure, $thisPartId = ''
) {
  switch ($thisPartStructure->type) {
    case $reqType:
      if ($thisPartId == '') { // means 'entire message' per RFC2060
        $thisPartId = '1'; // message has one part only here, and this
          // is the part number of the one and only part
      }
      $part = imap_fetchbody($mailbox, $msgNo, $thisPartId);
      $part = mime_decode($part, $thisPartStructure->encoding);
      if ($thisPartStructure->ifsubtype)
        $subtype = $thisPartStructure->subtype;
      else
        $subtype = 'unknown';
      break;
    case TYPEMULTIPART:
      $subPartPrefix = ($thisPartId == '' ? '': "$thisPartId.");
      foreach (
        $thisPartStructure->parts as $subPartIdx => $subPartStructure
      ) {
        $subPartId = $subPartPrefix.($subPartIdx+1);
        list($part, $subtype) = extract_typed_part(
          $mailbox, $msgNo, $reqType, $subPartStructure, $subPartId
        );
        if (isset($part)) break;
      }
      break;
  }
  return(array($part, $subtype));
}
```

```php
/** converts audio formats to AMR audio format if possible
  *
  * E-mails originating from MMS might have audio data transcoded to
  * WAV, which has to be undone (see
  * http://support.t-mobile.com/knowbase/root/public/tm21396.htm ). This
  * function can handle the MIME subtypes 'wav' and 'amr' but not 'x-wav'
  * etc..
  *
  * @param $msgAudioFile file name of the file to convert
  * @param $msgAudioSubtype MIME subtype of the file to convert
  * @return array with filename and subtype of the converted file; might
  *   be NULL if the conversion failed or no input was given; filename
  *   shares the same directory and basename with $msgAudioFile and has
  *   extension .amr; this file is overwritten if it exists
  */
function audio_to_amr($msgAudioFile, $msgAudioSubtype) {
  if (isset($msgAudioFile)) {
    // conversion necessary and possible?
    if (in_array($msgAudioSubtype, array('wav', 'WAV'))) {
      $dir = (
        (dirname($msgAudioFile) == '.' or dirname($msgAudioFile) == '') ?
          '' : (dirname($msgAudioFile) . '/')
      );
      $msgAudioFileWAV = $msgAudioFile;
      $msgAudioFileRAW = $dir.basename($msgAudioFileWAV, '.wav').'.raw';
      $msgAudioFileAMR = $dir.basename($msgAudioFileWAV, '.wav').'.amr';

      exec(
        SOX_CALL .
          " $msgAudioFileWAV -r 8000 -w -c 1 -s $msgAudioFileRAW 2>&1",
        $outputSox,
        $retValSox
      );
      if ($retValSox != 0) {
        error_log("WARNING: conversion aborted as sox had problems:\n"
          . implode("\n",$outputSox) . "\n", 3, LOGFILE_NAME);
        $msgAudioFile = $msgAudioSubtype = NULL;
      }
      else {
        exec(
          AMRENCODER_CALL .
            " MR122 $msgAudioFileRAW $msgAudioFileAMR 2>&1",
          $outputAmrEncoder,
          $retValAmrEncoder
        );
        if ($retValAmrEncoder != 0) {
          error_log(
            "WARNING: conversion aborted as amr-encoder had problems:\n"
            . implode("\n",$outputAmrEncoder) . "\n", 3, LOGFILE_NAME);
          $msgAudioFile = $msgAudioSubtype = NULL;
        }
        else {
          $msgAudioSubtype = 'amr';
          $msgAudioFile = $msgAudioFileAMR;
        }
      }
    }
    // conversion necessary but not possible?
```

```
    elseif ($msgAudioSubtype != 'amr' and $msgAudioSubtype != 'AMR') {
      error_log(
        "WARNING: audio format audio/$msgAudioSubtype cannot be " .
        "converted yet, ignoring audio data\n",
        3, LOGFILE_NAME
      );
      $msgAudioFile = $msgAudioSubtype = NULL;
    }
  }
  return(array($msgAudioFile, $msgAudioSubtype));
}

function upload($fileName, $mimeSubtype) {
  if (empty($fileName)) { // nothing to do
    return(array(NULL, NULL));
  }
  $conn = ftp_connect(UPLOAD_FTP_SERVER);
  if ($conn === false) {
    error_log("ERROR: unable to connect to " . UPLOAD_FTP_SERVER .
      "; aborting upload\n", 3, LOGFILE_NAME);
    return(array(NULL, NULL));
  }
  $success = ftp_login($conn, UPLOAD_FTP_USER, UPLOAD_FTP_PASS);
  if ($success === false) {
    error_log("ERROR: unable to login to " . UPLOAD_FTP_SERVER .
      "; aborting upload\n", 3, LOGFILE_NAME);
    return(array(NULL, NULL));
  }
  $success = ftp_put ($conn, basename($fileName), $fileName, FTP_BINARY);
  if ($success === false) {
    error_log("ERROR: unable to upload $fileName; aborting upload\n",
      3, LOGFILE_NAME);
    return(array(NULL, NULL));
  }
  ftp_close($conn);
  return(array(UPLOAD_BASE_URL .'/'. basename($fileName), $mimeSubtype));
}
```

B.4 Main program

```
//////////////////////////// MAIN PROGRAM ////////////////////////////

error_log(
  "NOTICE: ------- invoked on " . date('Y-m-d G:i:s') . " -------\n",
  3,
  LOGFILE_NAME
);

// connect to e-mail account
$mailbox = imap_open(
  // might use another name for default mailbox than correct, but works
  '{' . IMAP_SERVER .':' . IMAP_PORT . '/imap/notls}',
  IMAP_LOGIN,
  IMAP_PASS
);
if ($mailbox === false ) {
  error_log('ERROR: cannot connect to mailbox: '.imap_last_error()."\n",
    3, LOGFILE_NAME);
```

```
  exit();
}

// determine recent messages, sorted by date
// (they are recent up to the next imap_close())
$allMsgsByDate = imap_sort($mailbox, SORTDATE, 0);
$recentMsgs = imap_search ($mailbox, 'RECENT'); // for tests use 'ALL'
// imap_search() returns false for no results or error:
if ($recentMsgs === false) $recentMsgs = array();
$recentMsgsByDate = array_values( // array_values() reindexes an array
  array_intersect($allMsgsByDate, $recentMsgs)
);
error_log(
  "NOTICE: Number of messages to process: " . count($recentMsgs) . "\n",
  3,
  LOGFILE_NAME
);

// traverse recent e-mails, broadcasting them as MM
foreach ($recentMsgs as $recentMsgsKey => $msgNo) {

  // extract header information
  $msgUid = imap_uid($mailbox, $msgNo);
  $msgHeader = imap_headerinfo($mailbox, $msgNo);
  $msgSubject = $msgHeader->Subject;
  $msgDate = $msgHeader->Date;
  // because from is <MSISDN>@<domain>.<tld> :
  list($msgFromMSISDN, ) = explode('@', $msgHeader->from[0]->mailbox);
  $msgFromName = $participants[$msgFromMSISDN];
  $msgRecipients = $participants;
  unset($msgRecipients[$msgFromMSISDN]); // sender is no recipient

  // extract text content
  $msgStructure = imap_fetchstructure($mailbox, $msgNo);
  list($msgText, $msgTextSubtype) =
    extract_typed_part($mailbox, $msgNo, TYPETEXT, $msgStructure);
  $msgTextSubtype = strtolower($msgTextSubtype);

  // extract audio content, saving to amr audio file
  list($msgAudio, $msgAudioSubtype) =
    extract_typed_part($mailbox, $msgNo, TYPEAUDIO, $msgStructure);
  $msgAudioSubtype = strtolower($msgAudioSubtype);
  if (isset($msgAudio)) {
    $msgAudioFile = "$msgUid.$msgAudioSubtype";
    $msgAudioFileHandle = fopen($msgAudioFile, 'wb'); // overwrites files
    $success = fwrite($msgAudioFileHandle, $msgAudio);
    if ($success === false) {
      error_log(
        "WARNING: cannot write to $msgAudioFile, ignoring audio data\n",
        3,
        LOGFILE_NAME
      );
      unset($msgAudioFile);
    }
    fclose($msgAudioFileHandle);
  }
  list($msgAudioFile, $msgAudioSubtype) =
    audio_to_amr($msgAudioFile, $msgAudioSubtype);
```

```
  list($msgAudioFile, $msgAudioSubtype) =
    upload($msgAudioFile, $msgAudioSubtype);

  // compose a MM and send it to the group
  if (isset($msgText) or isset($msgAudioFile)) {
    $mmSubject = mm_subject_constraints("$msgFromName $msgSubject");
    $mm = new MultimediaMessage($mmSubject, $msgRecipients);
    $mm->addSlide(NULL, $msgText, $msgAudioFile);
    error_log(
      "NOTICE: going to send this message:\n". print_r($mm, true) ."\n",
      3,
      LOGFILE_NAME
    );
    $success = $mm->send();
    if ($success)
      error_log(
        "NOTICE: message '$mmSubject' " .
          "successfully sent to all recipients:\n" .
          print_r($mm->mmsLastResponses, true),
        3,
        LOGFILE_NAME
      );
    else
      error_log(
        "WARNING: message '$mmSubject' " .
          "could not be sent to all recipients:\n" .
          print_r($mm->mmsLastResponses, true),
        3,
        LOGFILE_NAME
      );
  }
  else {
    error_log(
      "NOTICE: message '$mmSubject' " .
        "not sent as it contains no text or audio body parts\n",
      3,
      LOGFILE_NAME
    );
  }
}

imap_close($mailbox);
?>
```

Index of glossary items

- 3G I
- Activity set I
- adaptability I
- affordance I
- Agile collection II
- agile design principles II
- Agile message II
- agile organizational paradigm II
- agile paradigm II
- agility II
- AMR III
- analytic model III
- awareness III
- behavior-setting III
- CGW III
- CGW:RP III
- collaboration IV
- combining pattern IV
- communication IV
- community IV
- community groupware V
- communityware V
- Connectable artifact V
- controlled vocabulary V
- Conversational message VI
- Conversation paradigm VI
- coordination VI
- coordinating representation VI
- CSCL VII
- CSCW VII
- CSCW model VII
- DSR VII
- environment paradigm VII
- ethnography VII
- extreme collaboration VIII
- framework VIII
- iCalendar VIII
- industrial engineering VIII
- inhabitants VIII
- Interacting artifact IX
- latest time of delivery IX
- lean production IX
- linear organizational paradigm IX
- linear paradigm X
- mandated adoption X
- MC³ X
- Message X
- milieu X
- MM X
- MM3 XI
- MM7 XI
- MMS XI
- model XI
- MSISDN XII
- MUD XII
- multimodal XII
- Negotiated synchrony XII
- network effect XII
- network community XIII
- obtrusiveness acceptance level XIII
- organizational design XIII
- organizational model XIII
- organizational paradigm XIII
- pattern XIV
- phatic communication XIV
- PLML XIV
- PoC XIV
- pragmatic action XIV

Index of glossary items

- PTT XV
- Push to Talk XV
- Push to Talk over Cellular XV
- real-time XV
- reified interaction XVI
- response ability XVI
- semantic navigation XVI
- sequential multimodal input XVI
- simplicity XVI
- SM XVII
- SMS XVII
- SMTP XVII
- SOAP XVII
- social navigation XVII
- sources of synomorphy XVII
- standing pattern of behavior XVII
- Subscription XVIII
- synthetic model XVIII
- SyncML XVIII
- sync point XVIII
- synomorph XVIII
- system XVIII
- task environment XIX
- third place XIX
- tool paradigm XIX
- typeless artifact XIX
- UI XIX
- Unbound communicator XX
- unit XX
- Universal member XX
- VAS XX
- VASP XX
- vCard XX
- VoIP XXI
- voluntary adoption XXI
- WAP XXI
- XC XXI
- XP XXI

Index of objects

- Object 1: mobile groupware for communities: research space, current focus and mapping to this thesis' structure **5**
- Object 2: different categories of collaborational models in context of each other **11**
- Object 3: placement of different model types in the system engineering process **13**
- Object 4: domain coverage of referenced sociological theories (navigational aid) **14**
- Object 5: correlation of behavior-setting and pattern concepts **18**
- Object 6: Manifesto for Agile Software Development, as shown in [Beck et al :a03] **23**
- Object 7: Agile design principles from the Agility Forum [Dove :a04] **26**
- Object 8: correlation of Agile Manifesto principles with agile design principles **26**
- Object 9: evaluating candidates for the organizational paradigm of CGW **51**
- Object 10: MC³ pattern overview diagram **62**

Bibliography

[3GPP :a01] 3rd Generation Partnership Project. *3GPP TS 23.140 V6.13.0 - Multimedia Messaging Service (MMS); Functional description; Stage 2 (Release 6)*. Valbonne, France. 2006. 3GPP Technical Specification. http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/23140-6d0.zip [accessed 2006-07-19]

[3GPP :a02] 3rd Generation Partnership Project. *3GPP TS 26.071 V6.0.0 (2004-12) - 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mandatory speech CODEC speech processing functions; AMR speech CODEC; General description (Release 6)*. Valbonne, France. 2004. 3GPP Technical Specification. http://www.3gpp.org/ftp/Specs/archive/26_series/26.071/26071-600.zip [accessed 2006-07-25]

[3GPP :a03] 3rd Generation Partnership Project. *3GPP TS 24.011 V6.1.0 - Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface (Release 6)*. Valbonne, France. 2005. 3GPP Technical Specification. http://www.3gpp.org/ftp/Specs/archive/24_series/24.011/24011-610.zip [accessed 2006-07-26]

[3GPP :a04] 3rd Generation Partnership Project. *3GPP TS 26.104 V 5.2.0 - ANSI-C code for the floating-point Adaptive Multi Rate (AMR) speech codec (Release 5)*. Valbonne, France. 2003. 3GPP Technical Specification. http://www.3gpp.org/ftp/Specs/archive/26_series/26.104/26104-520.zip [accessed 2006-07-20]

[ASV :a01] Schaff, Philip (ed.). *The American Standard Version of the Holy Bible* . Nashville, TN, USA. 1901. http://www.ebible.org/bible/asv/ [accessed 2006-06-19]

[Ackerman et al :a01] Ackerman, Mark S.; Hindus, Debby; Mainwaring, Scott; Starr, Brian. Thunderwire: A Field Study of an Audio-Only Media Space. In: Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'96). Boston, MA, USA. 1996: pp. 238-247. http://www.eecs.umich.edu/~ackerm/pub/96b21/cscw96.tw.pdf [accessed 2006-05-10]

[Adler et al :a01] Adler, Annette; Ito, Mizuko; Mynatt, Elizabeth D.; O'Day, Vicki L.. Design For Network Communities. In: Proceedings of CHI 97. Atlanta, GA, USA.

1997: pp. 210-217. http://www-static.cc.gatech.edu/~everyday-computing/publications/netcom-chi1997.pdf [accessed 2006-05-11]

[Adler et al :a02] Adler; Annette; Bellotti, Victoria; Dourish, Paul; Henderson Jr., D. Austin. *Your Place or Mine? Learning from Long-Term Use of Audio-Video Communication*. Computer Supported Cooperative Work (1996) 5 1: pp. 33-62. http://www.ics.uci.edu/~jpd/publications/1996/jcscw-office-share.pdf [accessed 2006-06-03]

[Agile Alliance :a01] Agile Alliance. *Agile Alliance*. 2006. http://www.agilealliance.org/ [accessed 2006-04-02]

[Agilemobile.com :a01] Agilemobile.com. *Agile Messenger*. 2004. http://www.agilemobile.com/agile_messenger.html [accessed 2006-07-24]

[Agility International :a01] Agility International. *Briefing on Agility and Business Agility*. 2002. http://www.agility.co.uk/ab1.html [accessed 2006-05-22]

[Agre :a01] Agre, Philip E.. *Rethinking Networks and Communities in a Wired Society*. Pasadena, CA, USA. 1999. Paper presented to the American Society for Information Science. http://polaris.gseis.ucla.edu/pagre/asis.html (a draft from 1999-12-10) [accessed 2006-05-11]

[Aguanno et al :a01] Kevin Aguanno (ed.). *Managing Agile Projects*. Oshawa, Ontario, Canada. 2005. http://www.agilesecrets.com/Introduction.pdf (pp. 17-52) [accessed 2006-06-26]

[Alterman et al :a02] Alterman, Richard; Garland, Andrew. *Convention in Joint Activity*. Cognitive Science (2001) 25 4: pp. 611-657. http://www.cs.brandeis.edu/~alterman/papers_pdf/convention.pdf [accessed 2006-04-08]

[Alterman et al :a03] Alterman, Richard; Feinman, Alex; Introne, Josh; Landsman, Seth. Coordinating Representations in Computer-Mediated Joint Activities. In: Proceedings of the 23rd Annual Conference of the Cognitive Science Society. Mahwah, NJ, USA. 2001: pp. 43-48. http://www.cs.brandeis.edu/~alterman/papers_pdf/cr.cogsci.pdf [accessed 2006-03-08]

[Alterman et al :a08] Alterman, Richard; Introne, Joshua. *Structuring Coordination to Provide Intelligent Assistance in Groupware*. Waltham, MA, USA. 2005. Technical Report. http://www.cs.brandeis.edu/~alterman/papers_pdf/adaptc.pdf [accessed 2006-04-10]

[Alterman et al :a10] Alterman, Richard; Carpenter, Tamitha; Zito-Wolf, Roland. *Pragmatic Action*. Cognitive Science (1998) 22 1: pp. 53-105. http://www.cs.brandeis.edu/~alterman/papers_pdf/pragmatic.action.pdf [accessed 2006-04-10]

[Augustine :a01] Augustine, Sanjiv. *Managing Agile Projects*. Indianapolis, IN, USA. 2005.

[Barker :a01] Barker, R. G.. *Ecological psychology: concepts and methods for studying the environment of human behavior*. Stanford, CA, USA. 1968.

[Bayley et al :a01] Bayley, Oliver; Hagström, Anna Elisabeth; Hindus, Debby; Leduc, Nicole; Mainwaring, Scott D.. *Casablanca: Designing Social Communication Devices for the Home*. Seattle, WA, USA. 2001. Proceedings of SIGCHI'01. http://www.debbyhindus.com/documents/Hindus-CHI01-color.pdf [accessed 2006-04-17]

[Beck :a02] Beck, Kent. *Extreme Programming Explained: Embrace Change*. Boston, MA, USA. 1999.

[Beck et al :a03] Beck, Kent; Beedle, Mike; van Bennekum, Arie; Cockburn, Alistair; Cunningham, Ward; Fowler, Martin; Grenning, James; Highsmith, Jim; Hunt, Andrew; Jeffries, Ron; Kern, Jon; Marick, Brian; Martin, Robert C.; Mellor, Steve; Schwaber, Ken; Sutherland, Jeff; Thomas, Dave. *Manifesto for Agile Software Development*. 2001. http://www.agilemanifesto.org/ [accessed 2006-04-02]

[Beck et al :a04] Beck, Kent; Beedle, Mike; van Bennekum, Arie; Cockburn, Alistair; Cunningham, Ward; Fowler, Martin; Grenning, James; Highsmith, Jim; Hunt, Andrew; Jeffries, Ron; Kern, Jon; Marick, Brian; Martin, Robert C.; Mellor, Steve; Schwaber, Ken; Sutherland, Jeff; Thomas, Dave. *Principles behind the Agile Manifesto*. 2001. http://www.agilemanifesto.org/principles.html [accessed 2006-05-22]

[Bellamine et al :a01] Bellamine, Narjès; Saoud, Ben. *Modeling and Simulating Extreme Collaboration: An Agent-Based Approach*. Chania, Crete. 2002. Research Paper presented at the COSI (Complexity in Social Sciences) Summer School 2002. http://www.irit.fr/COSI/summerschool/bsecoll.pdf [accessed 2006-06-20]

[Boehm et al :a01] Boehm, Barry; Turner, Richard. Observations on Balancing Discipline and Agility. In: Proceedings of the Agile Development Conference. Salt

Lake City, UT, USA. 2003: pp. 32-39. http://agiledevelopmentconference.com/2003/files/P4Paper.pdf [accessed 2006-06-04]

[Borges et al :a01] Borges, Marcos R. S.; Kirsch-Pinheiro, Manuele; Lima, José Valdeni. A Framework for Awareness Support in Groupware Systems. In: Proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design (CSCWD'2002). Rio de Janeiro, Brazil. 2002: pp. 13-18. http://www-lsr.imag.fr/users/Manuele.Kirsch-Pinheiro/kirsch-lima-borges_cscwd2002.pdf [fetched 2006-07-30]

[Brignull :a05] Brignull, Harry. *Understanding and Designing for the Voluntary Adoption of Community Displays*. Sussex, UK. 2005. http://www.ux-design.net/pdf_version/harry_brignull_phd_thesis.pdf [accessed 2006-04-17]

[Brignull et al :a01] Brignull, Harry; Rogers, Yvonne. *Computational offloading: Supporting distributed team working through visually augmenting verbal communication*. Boston. 2003. Proceedings of 25th Annual Meeting of Cognitive Science Society. http://group.cs.brandeis.edu/~burner/paper.pdf [accessed 2006-04-11]

[Brignull et al :a03] Brignull, Harry (ed.). *Dynamo: a communal multi-user surface that supports cooperative sharing & exchange of digital media*. 2004. http://www.ux-design.net/dynamo-interactive.com/ [accessed 2006-04-13]

[Brignull et al :a04] Brignull, H.; Izadi, S.; Rodden, Tom; Rogers, Yvonne; Underwood, M.. *Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media*. Sussex, UK. 2003. Proceedings of 16th annual ACM symposium on User interface software and technology (UIST), Vancouver, Canada. http://www.cogs.susx.ac.uk/interact/papers/pdfs/159izadi-435.pdf [accessed 2006-04-17]

[Brodbeck et al :a01] Brodbeck, Dominique; Chalmers, Matthew; Rodden, Kerry. *The Order of Things: Activity-Centred Information Access*. Computer Networks and ISDN Systems (1998) 30 1-7: pp. 359-367. http://www.dcs.gla.ac.uk/~matthew/papers/WWW7/www98.pdf [accessed 2006-06-13]

[Brynjolfsson et al :a01] Brynjolfsson, Erik; Mendelson, Haim. *Information Systems and the Organization of Modern Enterprise*. Journal of Organizational Computing and Electronic Commerce (1993) 3 3: pp. 245 - 256. http://ccs.mit.edu/papers/CCSWP200/ [accessed 2006-04-26]

[Brzozowski et al :a01] Brzozowski, Mike; Carattini, Kendra; Hu, Jiang; Klemmer, Scott R.; Mihelich, Patrick; Ng, Andrew Y.. *groupTime: Preference-Based Group Scheduling*. Montréal, Québec, Canada. 2006. To be published in Proceedings of CHI 2006. http://hci.stanford.edu/publications/2006/groupTimeCHI2006.pdf [accessed 2006-04-15]

[CPSR :a01] Computer Professionals For Social Responsibility. *Liberating Voices! - A Pattern Language for Communication Revolution*. 2006. http://trout.cpsr.org/program/sphere/patterns/ [accessed 2006-04-18]

[CSEG :a01] Cooperative Systems Engineering Group (CSEG) of the Lancaster University Computing Department. *Patterns of Interaction: a Pattern Language for CSCW - Patterns of Cooperative Interaction.* 2001. http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/patterns.html [accessed 2006-04-17]

[Carnegie Mellon University :a01] Carnegie Mellon University. *The CMU Sphinx Group Open Source Speech Recognition Engines*. 2006. http://cmusphinx.sourceforge.net/ [accessed 2006-04-03]

[Chachere et al :a01] Chachere, John; Kunz, John; Levitt, Raymond. *Can You Accelerate Your Project Using Extreme Collaboration? A Model Based Analysis*. Stanford, UK. 2003. CIFE Technical Report #154. http://www.stanford.edu/group/CIFE//online.publications/TR154.pdf [accessed 2006-06-20]

[Chalmers et al :a02] Chalmers, Matthew; Dourish, Paul. *Running Out of Space: Models of Information Navigation*. Glasgow, U.K.. 1994. Ancillary Proceedings of BCS HCI'94, short paper. http://www.dcs.gla.ac.uk/~matthew/papers/hci94.pdf [accessed 2006-06-14]

[Chan et al :a01] Chan, Christopher; Kessel, Angela. *Castaway: A Context-Aware Task Management System*. Montréal, Québec, Canada. 2006. To be published in Proceedings of CHI 2006. http://hci.stanford.edu/publications/2006/chi2006-wip-kessell.pdf [accessed 2006-04-15]

[Cheverst et al :a01] Cheverst, Keith; Dix, Alan; Gibbs, Martin; Graham, Connor; Rouncefield, Mark. *Understanding space, place and 'community'*. Rome, Italy. 2005. To appear in: Proceedings of Interact '05 workshop on 'Space, Place and

Experience in HCI'. http://www.caside.lancs.ac.uk/publications/Space_Place_draft.pdf [accessed 2006-04-19]

[Covi et al :a01] Covi, Lisa; Krishnan, M. S.; Olson, Judith S.; Teasley, Stephanie. *How does radical collocation help a team succeed?*. Philadelphia, Pennsylvania, United States. 2000. Proceedings of the 2000 ACM conference on Computer supported cooperative work. http://possibility.com/Misc/p339-teasley.pdf [accessed 2006-06-20]

[Crabtree :a01] Crabtree, Andy; Hemmings, Terry; Rodden, Tom. *Pattern-based Support for Interactive Design in Domestic Settings*. In: Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques. London, UK. 2002: pp. 265-276 http://www.equator.ac.uk/var/uploads/DIS_2002.pdf [accessed 2006-04-15]

[Crabtree :a08] Crabtree, Andy. *ACCORD Deliverable 1.3 - Patterns of Home Life*. EU Disappearing Computer Initiative (IST-2000-26364). 2001. Research Report. http://www.mrl.nott.ac.uk/~axc/patterns_of_home_life/patterns_homepage.htm [accessed 2006-04-17]

[Crabtree :a12] Crabtree, Andy. *The Social Organization of Communication in Domestic Settings*. Manchester, UK. 2003. Proceedings of the 2003 Conference of the International Institute of Ethnomethodology and Conversation Analysis. http://www.equator.ac.uk/var/uploads/Crabtree, A. (2003) The social organization of communication in the home.2003.pdf [accessed 2006-04-18]

[Crabtree :a13] Crabtree, Andy. *Designing Collaborative Systems : A Practical Guide to Ethnography*. Berlin, Germany. 2003.

[Crabtree et al :a02] Crabtree, Andy; Hemmings, Terry; Rodden, Tom; Schnädelbach, Holger. *Patterns of Technology Usage in the Home: Domestic Legacy and Design*. Nottingham, UK. 2001. Technical Report Equator-01-015, University of Nottingham, The School of Computer Science and IT. http://www.mrl.nott.ac.uk/~axc/documents/Patterns.pdf [fetched 2006-07-30]

[Crabtree et al :a04] Crabtree, Andy; Hemmings, Terry; Rodden, Tom. Domestic Legacy and Design. In: Proceedings of The 1st Equator Workshop on Ubiquitous Computing in Domestic Environments. Nottingham, UK. 2001: pp. 147-164. http://www.mrl.nott.ac.uk/~axc/equator_workshop/Crabtree.pdf [accessed 2006-04-15]

[Crabtree et al :a07] Crabtree, Andy; Hemmings, Terry; Rodden, Tom. *Supporting Communication Within Domestic Settings*. Irvine, CA, USA. 2002. Proceedings of the 2003 Home Oriented Informatics and Telematics Conference (HOIT 2003). http://www.mrl.nott.ac.uk/~axc/documents/HOIT_2003.pdf [accessed 2006-04-17]

[Crabtree et al :a11] Crabtree, Andy; Hemmings, Terry; Mariani, John; Rodden, Tom. *Informing the Development of Calendar Systems for Domestic Use*. Helsinki. 2003. To appear in: Proceedings of the 8th European Conference on Computer Supported Cooperative Work, pp. 119-138. http://ubicomp.org/ubicomp2005/calls/callworkshops.shtml#w7 [accessed 2006-04-18]

[Crowston :a02] Crowston, Kevin. *A Taxonomy Of Organizational Dependencies and Coordination Mechanisms*. Cambridge, MA, USA. 1994. MIT Center for Coordination Science Technical Report 174. http://ccs.mit.edu/papers/CCSWP174.html [accessed 2006-04-26]

[Crowston et al :a01] Crowston, Kevin; Malone, Thomas W.. *The Interdisciplinary Study of Coordination*. ACM Computing Surveys (1994) 26 1: pp. 87-119. http://ccs.mit.edu/papers/CCSWP157.html [accessed 2006-07-25]

[Cunningham et al :a01] Cunningham, Ward. *Wiki History*. 2006. http://c2.com/cgi/wiki?WikiHistory [accessed 2006-06-22]

[Dawson et al :a01] Dawson, F.; Howes, T.; Smith, M.. *A MIME Content-Type for Directory Information*. Reston, VA, USA; Geneva, Switzerland. 1998. Request for Comments 2425. http://www.ietf.org/rfc/rfc2425.txt [accessed 2006-07-26]

[Dawson et al :a02] Dawson, F.; Howes, T.. *vCard MIME Directory Profile*. Reston, VA, USA; Geneva, Switzerland. 1998. Request for Comments 2426. http://www.ietf.org/rfc/rfc2426.txt [accessed 2006-07-26]

[Dewsbury et al :a01] Dewsbury, Guy; Rouncefield, Mark; Clarke, Karen; Sommerville, Ian. *Designing Appropriate Assistive Technology for Home Users: Developing Dependable Networks*. Rome, Italy. 2002. Proceedings of the CIB Working Group W084 - Building Non-Handicapping Environments "Inclusive Design and Mobility Response in Indoor/Outdoor Public Buildings and Facilities". http://www.dirc.org.uk/publications/inproceedings/papers/54.pdf [accessed 2006-04-15]

[Dourish :a02] Dourish, Paul. *The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents*. Computer-Supported Cooperative Work (2003) 12 Special Issue on Evolving Use of Groupware: pp. 465-490. http://www.ics.uci.edu/~jpd/publications/2002/jcscw-appropriation.pdf [accessed 2006-05-19]

[Dove :a01] Dove, Rick. *Enterprise Agility—What Is It?*. 2004. http://www.parshift.com/Files/Essays/Essay064.pdf [accessed 2006-05-22]

[Dove :a02] Dove, Rick. *Tools for Analyzing and Constructing Agility*. Bethlehem, PA, USA. 1994. Republished by Agility Forum as PA96-01 in Jan 1996. http://www.parshift.com/Files/PsiDocs/Rkd4Art4.pdf [accessed 2006-05-22]

[Dove :a03] Dove, Rick. Fundamental Principles for Agile Systems Engineering. In: Proceedings of the Third Annual Conference on Systems Engineering Research (CSER-05). Hoboken, NJ, USA. 2005: n.a.. http://www.parshift.com/Files/PsiDocs/Rkd050324CserPaper.pdf [accessed 2006-05-23]

[Dove :a04] Dove, Rick K.. Design Principles for Highly Adaptable Business Systems, With Tangible Manufacturing Examples. In: *Maynard's Industrial Engineering Handbook*. Ed.: Zandin, Kjell B.; Maynard, Harold B.. 5. edition. New York, NY, USA; et al.. 2001. pp. 9.3-9.26. http://www.parshift.com/Files/PsiDocs/ Rkd8Art3.pdf [accessed 2006-05-23]

[Dugdale et al :a01] Dugdale, Julie; Pavard, Bernard. *The contribution of complexity theory to the study of sociotechnical cooperative systems*. InterJournal: New England Complex Institute Electronic Journal (2000) n.a. manuscript number 335: 1-8. http://www.irit.fr/COSI/summerschool/bpcompcontrib.pdf [accessed 2006-06-20]

[E-Plus Service :a01] E-Plus Service GmbH & Co. KG. *E-Plus Service GmbH & Co. KG:* Besondere Bedingungen und Preisliste der Zusatzdienstleistung „E-Plus MultimediaMessagingService“ („MMS“), gültig ab dem 01.04.2006. 2006. http://www.eplus.de/agb/down/preisliste_mms.pdf [accessed 2006-07-21]

[FH Giessen-Friedberg :a01] Vogel, Christian (ed.). Prüfungsordnung des Fachbereiches Mathematik, Naturwissenschaften und Informatik der Fachhochschule Gießen-Friedberg für den Diplom-Studiengang Informatik vom 24.10.2001. 2001. http://www.fh-giessen.de/fachbereich/mni/download/DiplomPO.pdf [fetched 2006-07-30]

[Finan et al :a01] Finan, Robert; Niklfeld, Georg; Pucher, Michael. Component-based multimodal dialog interfaces for mobile knowledge creation. In: Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management. Toulouse, France. 2001: 103-110. http://acl.ldc.upenn.edu/W/W01/W01-1016.pdf [accessed 2006-04-07]

[Fincher :a01] Fincher, Sally. *HCI Pattern-Form Gallery*. 2006. http://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html [accessed 2006-04-17]

[Fincher :a02] Fincher, Sally. *CHI 2003 Workshop Report - Perspectives on HCI Patterns: Concepts and tools (introducing PLML)*. Interfaces (2003) 33 56: pp. 26-28. http://www.bcs-hci.org.uk/interfaces/interfaces56.pdf [accessed 2006-05-08]

[Five Across Inc :a01] Five Across, Inc. *Five Across Collaborate*. 2005. http://www.fiveacross.com/products/collaborate.shtml [accessed 2006-07-24]

[Frank :a01] Frank, Aiko. Das ASCEND-Modell zur Unterstützung kooperativer Prozesse. Stuttgart. 2002. http://deposit.ddb.de/cgi-bin/dokserv?idn=966105281&dok_var=d1&dok_ext=pdf&filename=966105281.pdf [accessed 2006-05-05]

[Fuchs et al :a01] Fuchs, Ludwin; Mark, Gloria; Sohlenkamp, Markus. Supporting Groupware Conventions through Contextual Awareness. In: Proceedings of ECSCW'97. Lancaster, UK. 1997: pp. 253-268. http://www.ics.uci.edu/~gmark/ECSCW97.pdf [accessed 2006-04-21]

[Funambol Inc :a01] Funambol, Inc.. *Funambol Project (formerly known as Sync4j)*. 2006. http://www.funambol.com/opensource/ [accessed 2006-04-07]

[Gamma et al :a01] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, MA, USA. 1995.

[Gensel et al :a01] Gensel, J.; Kirsch-Pinheiro, M.; Martin, H.. *Awareness on Mobile Groupware Systems (Draft)*. Berlin, Germany. 2004. LNCS 3284 - MATA 2004, 1st International Workshop on Mobility Aware Technologies and Applications. http://www-lsr.imag.fr/users/Manuele.Kirsch-Pinheiro/Draft_MATA04_KirschGenselMartin.pdf [accessed 2006-04-07]

[goyya.com :a01] goyya.com OHG. *Schnittstellenbeschreibung MMS-Gateway*. 2004. http://www.massenversand.de/mv2004/downloads/Interface_MMS.pdf [accessed 2006-07-14]

[goyya.com :a02] goyya.com OHG. *Preisliste MMS Inbound*. 2006. http://www.massenversand.de/mv2004/downloads/Preisliste_MMSInbound.pdf [accessed 2006-07-21]

[Halloran et al :a01] Halloran, John; Rogers, Yvonne; Rodden, Tom; Taylor, Ian. Creating new user experiences to enhance collaboration. In: Proceedings of INTERACT'03. Zurich. 2003: pp. 479-486. http://www.slis.indiana.edu/faculty/yrogers/papers/Halloran_interact03.pdf [accessed 2006-04-13]

[Halloran et al :a02] Halloran, John; Rogers, Yvonne; Scaife, Mike. Taking The 'No' Out Of Lotus Notes: Activity Theory, Groupware, and Student Groupwork. In: Proceedings of Computer Supported Collaborative Learning 2002. Denver, CO, USA. 2002: pp. 169-178. http://www.slis.indiana.edu/faculty/yrogers/papers/CSCL02.pdf [accessed 2006-04-13]

[Heise Zeitschriften Verlag :a01] Heise Zeitschriften Verlag. *In Deutschland steigt die Zahl der MMS-Botschaften*. 2005. http://www.heise.de/mobil/newsticker/meldung/62419 [accessed 2006-07-24]

[Hellkvist :a01] Hellkvist, Stefan. *MMSLIB*. 2006. http://hellkvist.org/software/#MMSLIB [accessed 2006-07-21]

[Highsmith :a02] Highsmith, Jim. *History: The Agile Manifesto*. 2001. http://www.agilemanifesto.org/history.html [accessed 2006-07-24]

[Highsmith:a01] Highsmith, Jim. *Agile Project Management : Creating Innovative Products*. Boston, MA, USA. 2004.

[Hindus :a01] Hindus, Debby. The Importance of Homes in Technology Research. In: Proceedings of the Second International Workshop on Cooperative Buildings (CoBuild'99). London, UK. 1999: pp. 199-207. http://www.debbyhindus.com/documents/Hindus-CoBuild99.pdf [accessed 2006-05-10]

[Howell et al :a01] Howell, Greg; Koskela, Lauri. *The Theory of Project Management: explanation to novel methods*. Gramado, Brazil. 2002. Proceedings of the 10th Annual Conference on Lean Construction (IGLC-10).

http://www.cpgec.ufrgs.br/norie/iglc10/papers/47-Koskela&Howell.pdf [accessed 2006-06-05]

[Hughes et al :a01] Hughes, John; O'Brien, Jon; Rodden, Tom; Rouncefield, Mark; Viller, Stephen. *Patterns of Home Life: Informing Design For Domestic Environments*. Personal Technologies (2000) 4 1: pp. 25-38. http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/reports/PersonalTech.pdf [accessed 2006-04-15]

[Hutchins :a01] Hutchins, Edwin. *Cognition in the Wild*. Cambridge, MA, USA. 1995.

[Höök :a01] Höök, Kristina. Social navigation: from the web to the mobile . In: *Mensch & Computer 2003: Interaktion in Bewegung* . Ed.: Szwillus, G.; Ziegler, J.. Stuttgart, Germany. 2003: pp. 17-20. http://mc.informatik.uni-hamburg.de/konferenzbaende/mc2003/konferenzband/muc2003-01-hoeoek.pdf [accessed 2006-06-13]

[Jarke :a01] Jarke, Matthias. Mensch-Computer-Interaktion: Fortschritte und Perspektiven. In: *Fraunhofer-Gesellschaft. Jahresbericht 2004* . Ed.: Thum, M.. Munich, Germany. 2005: pp. 44-51. http://publica.fhg.de/eprints/N-28535.pdf [accessed 2006-04-21]

[Kvale et al:a01] Kvale, Knut; Warakagoda, Narada Dilp; Knudsen, Jan Eikeset. *Speech Centric Multimodal Interfaces for Mobile Communication Systems*. Telektronikk (2003) 99 2: pp. 104-117. http://www.eurescom.de/~public-web-deliverables/P1100-series/P1104/Multimodal_Interfaces_104_117.pdf [accessed 2006-04-05]

[Liebowitz et al :a01] Liebowitz, S.; Margolis, S.. Network Externalities (Effects). In: *The New Palgrave Dictionary of Economics and the Law* . Ed.: Eatwell, J.; Milgate, M.; Newman, P. London, UK. 1998: pp. n.a.. http://www.utdallas.edu/~liebowit/palgrave/network.html [accessed 2006-05-21]

[Lo et al :a01] Lo, Tin-Hang; Meng, Helen M.. *WAP-Speech: Deriving Synergy between WAP and the Spoken Dialog Interface*. Honolulu, Hawaii, USA. 2002. Proceedings of the Eleventh International World Wide Web Conference. http://www2002.org/CDROM/poster/81.pdf [accessed 2006-04-04]

[Mark :a01] Mark, Gloria. *Extreme Collaboration*. Communications of the ACM (2002) 45 6: pp. 89-93. http://www.ics.uci.edu/~gmark/Mark-CACM-final.pdf [accessed 2006-03-26]

[Mark :a02] Mark, Gloria. *Conventions and Commitments in Distributed CSCW Groups*. Computer Supported Cooperative Work: The Journal of Collaborative Computing (2002) 11 3-4: pp. 349-387. http://www1.ics.uci.edu/~gmark/CSCW-final.pdf [accessed 2006-04-21]

[mbuni.org :a01] mbuni.org. *Mbuni: Open Source MMS Gateway*. 2006. http://www.mbuni.org/ [accessed 2006-07-21]

[Munkes :a01] Munkes, Jörg. Sozialer Vergleich bei der computergestützten Gruppenarbeit. Tübingen, Germany. 2002. http://deposit.ddb.de/cgi-bin/dokserv?idn=97416562x&dok_var=d1&dok_ext=pdf&filename=97416562x.pdf [accessed 2006-05-05]

[NISO :a01] National Information Standards Organization. *ANSI/NISO Z39.19-2005: Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. Bethesda, MD, USA. 2005. http://www.niso.org/standards/resources/Z39-19-2005.pdf [accessed 2006-07-26]

[Nardi et al :a01] Nardi, Bonnie A.; Bradner, Erin; Whittaker, Steve. *Interaction and Outeraction: Instant Messaging in Action*. Philadelphia, PA. 2000. Proceedings of CSCW'00. http://www.darrouzet-nardi.net/bonnie/pdf/Nardi_outeraction.pdf [accessed 2006-05-11]

[Newell et al :a01] Newell, A.; Simon, H.. *Human problem solving*. Englewood Cliffs, NJ, USA. 1972.

[Nokia :a01] Nokia. *Nokia Xpress audio messaging - a personal new way to communicate*. 2005. http://press.nokia.com/PR/200502/979965_5.html [accessed 2006-07-25]

[Nokia UK Limited :a01] Nokia UK Limited. *Can I update my phone's firmware?*. 2006. http://www.nokia.co.uk/nokia/0,,77008,00.html#2 [accessed 2006-07-25]

[Norman :a02] Norman, Donald A.. *Emotion & Design: Attractive things work better*. Interactions Magazine (2002) 9 4: 36-42. http://www.jnd.org/dn.mss/Norman-EmotionAndDesign-InteractionsMag.pdf [accessed 2006-05-03]

[Norman :a05] Norman, Donald A.. *Minimizing the annoyance of the mobile phone*. Schaumburg, IL, Illinois. 2005. Paper prepared for the 2005 Motorola Research Visionary Board. http://www.jnd.org/dn.mss/minimizing_the.html [accessed 2006-05-03]

[Norman :a11] Norman, Donald. *Affordance, Conventions and Design*. Interactions (1999) 6 3: pp. 38-43. http://www.jnd.org/dn.mss/affordance_conv.html [accessed 2006-07-25]

[O2 Germany :a02] O_2 (Germany) GmbH & Co. OHG. *MMS to E-Mail*. 2006. http://www.o2online.de/o2/kunden/mobilemedia/handyguide/mmsdienste/mmstoemail/index.html [accessed 2006-07-21]

[Oldenburg :a01] Oldenburg, Ray. *The Great Good Place: Cafes, Coffee Shops, Bookstores, Bars, Hair Salons, and Other Hangouts at the Heart of a Community* . New York, NY, USA. 1989.

[Pankoke-Babatz :a01] Pankoke-Babatz, Uta. Designkonzept für Systeme zur computergestützten Zusammenarbeit unter Nutzung der Behavior-Setting-Theorie. Aachen. 2003. http://publica.fhg.de/eprints/B-87480.pdf [accessed 2006-04-21]

[Paradigm Shift :a01] Paradigm Shift International. *Resource and Forum for Agility, Knowledge Management, and Change Management*. 2006. http://www.parshift.com/ [accessed 2006-04-03]

[Pavard :a01] Pavard, Bernard. *Complexity Paradigm as a framework for the study of Cooperative Systems*. Chania, Crete, Greece. 2002. Research Paper presented at the at the COSI (Complexity in Social Sciences) Summer School 2002. http://www.irit.fr/COSI/summerschool/bpstudy.pdf [accessed 2006-06-20]

[Pearce :a01] Pearce, David. *Enabling Speech & Multimodal Services on Mobile Devices: The ETSI Aurora DSR standards & 3GPP Speech Enabled Services*. 2004. http://www.voicexmlreview.org/nov2004/features/dsr.html [accessed 2006-04-04]

[Rogers :a02] Rogers, Yvonne. Distributed Cognition and Communication. In: *The Encyclopedia of Language and Linguistics* . Ed.: Elsevier, Keith Brown. 2. edition. Oxford, UK.2005. pp. pp. 731-733. http://www.slis.indiana.edu/faculty/yrogers/papers/Rogers_DCog04.pdf [accessed 2006-04-13]

[STQ AURORA :a01] STQ AURORA. *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms - Doc. Nb. ES 201 108 Ver. 1.1.3*. . 2003. http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp?WKI_ID=18820 [accessed 2006-04-04]

[STQ AURORA :a02] STQ AURORA. *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms - Doc. Nb. ES 202 050 Ver. 1.1.4 Ref. RES/STQ-00083a*. 2005. http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp?WKI_ID=23463 [accessed 2006-04-04]

[Schwickert et al :01] Schwickert, Axel C.; Rey, Louis-Ferdinand. *Manuelle und elektronische Vorgangssteuerung*. 1996. http://geb.uni-giessen.de/geb/volltexte/2004/1704/pdf/Apap_WI_1996_05.pdf [accessed 2006-03-26]

[Schümmer :a03] Schümmer, Till. *Evolving a Groupware Pattern Language*. Helsinki, Finland. 2003. Position Paper for the ECSCW2003 Workshop "From Good Practices to Patterns". http://wwwpi6.fernuni-hagen.de:8080/gw-patterns/uploads/69/schuemmer.pdf [accessed 2006-04-17]

[Siemens AG :a01] Siemens AG. *White Paper - Push to Talk over Cellular*. . 2004. http://www.benqmobile.com/repository/211/21193/Push_to_talk_over_Cellular_e.pdf [accessed 2006-07-24]

[Smith-Berndtsson et al :a01] Smith-Berndtsson, Daniel; Åström, Jonas. *Is there a need for mobile services? - understanding the everyday lives of "ordinary" users with cars*. Ulvik in Hardanger, Norway. 2001. Proceedings of 24th Information Systems Research Seminar. http://www.viktoria.se/results/result_files/189.pdf [accessed 2006-04-15]

[Steiner :a01] Steiner, Ivan Dale. *Group Process and Productivity*. New York, NY, USA. 1972.

[Suchman :a01] Suchman, Lucy Alice. *Plans and situated actions : the problem of humanmachine communication*. Cambridge. 1987.

[teltarif.de :a01] teltarif.de Onlineverlag GmbH. *Multimedia-Messaging - MMS: Die Preise der Netzbetreiber*. . 2006. http://www.teltarif.de/i/mms.html?page=3 [accessed 2006-07-21]

[teltarif.de :a02] teltarif.de Onlineverlag GmbH. Die GPRS- und UMTS-Tarife im Überblick. . 2006. http://www.teltarif.de/i/gprs-kosten.html [accessed 2006-07-21]

[T-Mobile :a01] T-Mobile USA, Inc.. *Multimedia Messaging Service (MMS)*. . 2006. http://support.t-mobile.com/knowbase/root/public/tm21396.htm [accessed 2006-07-21]

[The Internet Society :a01] The Internet Society. *Mapping Between the Multimedia Messaging Service (MMS) and Internet Mail*. Reston, VA, USA; Geneva, Switzerland. 2006. Request for Comments 4356. http://www.ietf.org/rfc/rfc4356.txt [accessed 2006-07-21]

[The Internet Society :a02] The Internet Society. *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. Reston, VA, USA; Geneva, Switzerland. 1998. Request for Comments 2445. http://www.ietf.org/rfc/rfc2445.txt [accessed 2006-07-25]

[van der Veer :a01] van der Veer, Gerrit C.; van Welie, Martijn. *Pattern Languages in Interaction Design: Structure and Organization*. Zürich, Switzerland. 2003. Proceedings of Interact 2003. http://www.welie.com/papers/Welie-Interact2003.pdf [accessed 2006-04-17]

[Venkatesh :a01] Venkatesh, Alladi. *The Home of the Future: An Ethnographic Study of New Information Technologies in the Home*. Advances in Consumer Research (2001) 28: pp. 88-96. http://www.crito.uci.edu/NOAH/paper/HOF-Ethno.pdf [accessed 2006-05-10]

[Vlissides :a02] Vlissides, John. *Patterns: The Top Ten Misconceptions*. Object Magazine (1997) 7 1: pp. 30-33. http://www.research.ibm.com/designpatterns/pubs/top10misc.html [accessed 2006-05-08]

[Vodafone :a01] Vodafone D2 GmbH. Vodafone-InfoDok Nummer 338 - Fotos von Handy zu handy verschicken: viel Spaß mit Vodafone-MultimediaMessagingService (MMS). . 2006. http://www.vodafone.de/infofaxe/338.pdf [accessed 2006-07-21]

[Voicemate Inc :a02] Voicemate Inc.. *Voicemate TotalReach (SM)*. 2003. http://voicemate.com/brochures/TotalReach.pdf [accessed 2006-04-07]

[Weiser :a01] Weiser, Mark. *The Computer for the 21st Century*. Scientific American (1991) 265 9: pp. 94-104.

http://www.andrew.cmu.edu/user/cfchen/readings/pvc/computer-for-21-century.pdf [accessed 2006-04-18]

[Whatever Mobile :a01] Whatever Mobile GmbH. *Produktblatt MMS Versand V1.3*. 2006. http://www.whatevermobile.com/files/Produktblatt MMS Versand.pdf [accessed 2006-07-21]

[Whatever Mobile :a02] Whatever Mobile GmbH. *Produktblatt MMS Empfang V1.3*. 2006. http://www.whatevermobile.com/files/Produktblatt MMS Empfang.pdf [accessed 2006-07-21]

[Xu :a01] Xu, Chengmao. *Interaction and Collaboration Mechanisms for Distributed Communities and Groups in Educational Settings*. Munich, Germany. 2000. http://deposit.ddb.de/cgi-bin/dokserv?idn=962069604&dok_var=d1&dok_ext=pdf&filename=962069604.pdf [accessed 2006-05-05]