Chapter 7
Efficacious groupware development through Empirical Modelling

In the previous chapters, I pointed out the difficulties in developing groupware efficaciously and the issues surrounding the notion of participation in groupware development. In chapter 2, I argue that groupware must be able to evolve along with the work practices and human activities that it supports (cf. §2.4). Unfortunately, groupware is not trivial to develop due to its socio-technical nature. Even worse, the development of groupware is highly situated due to the fact that the group dynamics in each group are unlikely to be the same: this poses serious challenges to groupware developers (cf. §2.2). Consequently, groupware development often incorporates ethnographical and participatory design (PD) techniques in order to understand how people actually work in a particular situation. To increase the likelihood of groupware acceptance and to ease the adoption of groupware, user participation[139] has been thought as a remedy. However, the user involvement in the development process of any systems development, including groupware development, can sometimes be a tricky business (cf. 2.5.2). As Gasson (2003) observed, the nature of the development – its technological-oriented[140] focus – may remain the same even when practising a PD approach. In this respect, the culture in groupware development remains more or less user-centred or group-centred, but is not human-centred in the sense that I

[139] The terms "user" and "user participation" are far from perfect. As I have argued in chapter 3 (by citing various sources), the choice of these words hints at a technological focus. It may also mislead the reader into supposing that a "user" is technology-blind and has no technical competence.

[140] The technology-oriented focus can also be understood as a developer-oriented (or developer-driven) approach to systems development.

have elaborated earlier in this thesis (cf. §2.5.1 and §3.2).

As I have pointed out in chapter 3, the main issue behind participatory approaches is how we view the conception of roles and participation in the development process. If we set hard boundaries for the roles that the participants may play in the development process, any role-shifting behaviour becomes irrational. This not only affects the potential actions that the participants may carry out in the process, but also potentially inhibits advances in research into human-centred development. Even though this issue arises in contexts other than groupware development, it is the main obstacle to moving towards *efficacious groupware development* – a holistic perspective on groupware development (and evolution) that I set out in chapter 2 (cf. §2.5). Consequently, I argue for an alternative conceptual framework that may potentially be better aligned with the vision for efficacious groupware development.

The idea of practising an EM approach in the context of groupware development was inspired by various potential applications of EM, namely, requirements engineering (Sun, 1999), business process reengineering (Chen, 2001), financial sector (Maad 2002), and ubiquitous systems development (Wong, 2003). Throughout chapter 4, 5 and 6, I have discussed Empirical Modelling (EM) and examined its potential for collaborative modelling. So far in this thesis, the discussion of EM has been not directly connected with groupware development and the issue of participation. In this chapter, I argue that EM potentially facilitates efficacious groupware development.

In section 7.1, I argue that the core activity of groupware development can be thought of as a collaborative modelling process. This is to say, a shared understanding of the group activity is co-constructed through the construction of a model, and that model (which consists of shared pieces and private pieces) will gradually evolve into a stable artifact that can be regarded as a groupware. In section 7.2, I examine how EM may address the issues in participatory development (cf. chapter 3) which present the main obstacle in moving towards efficacious groupware development. Based on this perspective, I synthesize a conceptual framework – GroupPIE – for an EM approach to efficacious groupware development in section 7.3. Like other human-centred approaches, this framework is participatory. The framework facilitates participants with different backgrounds, expertise

and interests to co-construct and co-evolve the groupware and to co-evolve their understanding within a shared information space during the lifespan of the groupware from conception to disposal. In section 7.4 I discuss the limitations of the GroupPIE framework. In section 7.5, I argue that such a framework potentially moves towards "efficacious groupware development".

7.1 Groupware development as collaborative modelling

In chapter 5, I defined *collaborative modelling* (CM) as *the collaborative process of constructing an interactive computer-based model of a software system*. Viewed from the CM perspective, the core activity in groupware development is a two-fold collaborative modelling process: co-constructing a shared understanding of the group activity through the co-construction of a model consisting of shared pieces and private pieces that will gradually evolve into a stable artifact that can be regarded as groupware. In fact, such embodiment of the shared understanding of the group activity in the development process of groupware is consonant with Naur's (Naur, 1985a) account of the co-evolution between the developer's understanding and the construction of a program (cf. §3.1).

As with the *collaborative modelling* I described in chapter 5, the development process of groupware does not stay within the conceptual scope of one particular form of collaboration. Its construction involves different degrees of engagement, different relationships between the modeller and the agents, different modes of interaction in different situations and at different times. In relation to the construction of "models", the groupware development process generates two "models": i) a *conceptual model* of the group activity (and work practices), and ii) a *computer-based model* of the groupware. In this way, the construction of the latter model would lead the stakeholders to a better understanding of the group activity (i.e. the former model) and of how the groupware may fit into the context of that group activity.

It is worth noting that the groupware development as collaborative modelling (GD-as-CM) is a holistic view of the collaboration process in groupware development. It is not associated with, nor does it preclude, any particular form of participation. With this orientation, it is plausible to practise an EM approach to groupware development by practising an EM

approach to collaborative modelling (cf. chapter 5). In the next section, based on the discussion in section 2.5 and with EM in mind, I sketch a plausible process model for efficacious groupware development.

7.2 EM for human-centred development

In section 2.5, I argued that a human-centred focus is a pre-requisite for efficacious groupware development. In this section, I revisit human-centred development (HCD) through the lens of Empirical Modelling (EM) – an alternative approach to computing – that I introduced and discussed throughout chapter 4, 5 and 6. As mentioned in section 2.5, user participation is seen as a remedy for reducing the tensions among all stakeholders, particularly between the technical and the non-technical participants in the development, and as a prerequisite for establishing a suitable context for practising HCD. With this goal of HCD in mind, EM seems to offer a condition for such a high degree of participation to take place (cf. §7.2.1). In fact, there are many common characteristics between EM and PD (cf. §8.2), e.g. how they handle changes, diverse perspectives, etc throughout a systems development project. Apart from the potential for genuine participation, EM also has the potential to address three other HCD concerns which are particularly relevant in the context of groupware development: i) integrating the human and the technological (cf. §7.2.2), ii) enabling flexibility and evolution (cf. §7.2.3), iii) supporting diverse interaction and communication (cf. §7.2.4)[141].

[141] Beynon and Chan (2006, 2009) had already explored these three concerns in some detail in the context of Distributed Participatory Design (DPD). The discussion here is more oriented to the context of groupware development.

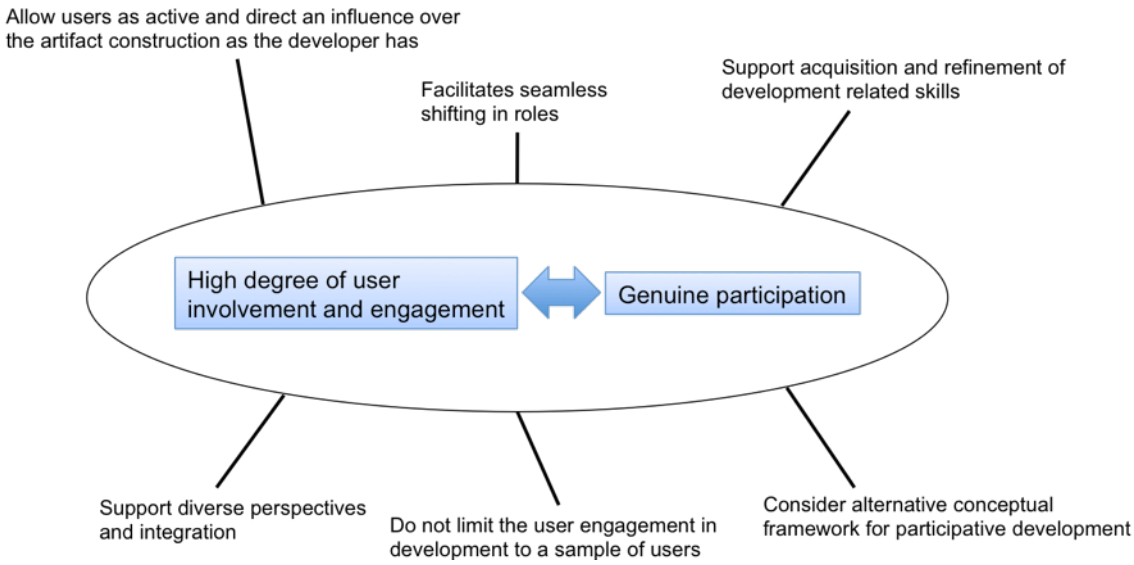


Figure 7.1 – Factors surrounding participatory development

7.2.1 EM for genuine participation

Earlier in this thesis, I made seven suggestions to address the participation problem in human-centred groupware development (cf. §3.5). As figure 7.1 illustrates, the intention behind these suggestions is to find a balance between a high degree of user involvement and engagement and the notion of *genuine participation* for the group of practitioners who will eventually use the groupware. In the following, I argue that EM potentially addresses most of these suggestions and is therefore potentially well-suited for human-centred groupware development.

Consider an alternative conceptual framework for participatory development

As I discussed throughout chapter 4, EM is fundamentally different from the traditional systems development approach (e.g. such as is described in Sommerville 2007). EM promotes modelling through observation, experimentation, and immediate experience. In relation to model construction, the EM process can be thought of as construing through identifying observables, dependencies, agents, and agencies (ODA). In contrast to a traditional systems development approach, EM emphasizes the construction of an understanding of a situation (and the activity) through and throughout the construction of the model. Due to this focus in the modelling process, an EM model (or construal) should not be considered as a complete end product in the traditional sense of systems development.

Allow users as active and direct an influence over the artifact construction as the developer has

As I argued in chapter 2 and 3, groupware may be resisted if its use disrupts the normal operation of the group. If a result of groupware development is a system that is based heavily on what the developers (including the ethnographers) can observe and interpret, but not on the future users' perspectives, the future users may be reluctant to adopt and adapt to the groupware. To make groupware and its users co-adapt to each other efficiently and effectively, it is important that the users of the developing groupware have as active and direct an influence as the developers.

Modelling with definitive scripts in EM potentially addresses this concern by blurring the boundaries of development and use, and the roles of developers and users. As I discussed in chapter 4, the semantic meaning of a (re)definition to the EM model is entirely at the modeller's discretion (also cf. Beynon, 2008). Take the jugs model (EMPA:jugsBeynon2008) as an example. There may be no syntactic difference between two semantically different redefinitions. For instance, the definition "contentA = 0" could be the result of a user's interaction "emptying jugA", but it could also be a developer's action "initialising a condition for jugA". Both developer and non-developers share a common interface: they can both make and trigger the same redefinition through the input window and through interaction with the "buttons" (i.e. the SCOUT windows) on the screen (cf. figure 4.7). That is to say, the state-changing effects within the model associated with these two different species of interaction ('development' and 'use') are expressed in one and the same space[142]. This affords an unusually literal meaning to the idea that the developer and the user interact in a shared modelling space when practising an EM approach to collaborative modelling (cf. chapter 5 and 6).

Taking all the above into account, EM not only blurs the boundary between developers and users, but also facilitates collaboration between the developers and the users in shaping the patterns of interaction among agents within the model. In other words, EM potentially

[142] Contrast the distinction between editing a line of code in a procedural program and executing the recipe that the programming code prescribes.

facilitates active and direct influence by the users of the groupware.

Support seamless shifting of roles

There is evidence that participants in two of the collaborative modelling case studies shifted their roles (viz. learner, user, developer, demonstrator/teacher) seamlessly from time to time during the collaborative modelling process (cf. §6.2 and §6.3). According to the video footage and the interaction history (i.e. part of the definitive scripts), this seamlessness of change may potentially be due to the instant feedback of the EM tool support – which is also a characteristic of EM (cf. §4.3). The role shifting phenomena almost went unnoticed both by the researcher and the modellers themselves (i.e. the participants in the case studies) because the modellers have to *do* virtually nothing in order to shift their roles in the collaborative modelling process.

In fact, such seamless shifting in roles may be viewed as a conflation of roles such as I discussed in chapter 3 (cf. figure 3.4 on page 90 and §3.4.3) if we step back and view the modeller as a human being who is interacting with the model with her native knowledge and skills rather than playing preconceived roles in the modelling (or development) process (cf. §3.2 and §3.4).

Support acquisition and refinement of development related skills

In the collaborative modelling case studies (cf. §6.2, §6.3, and §6.4), there are indications that the modellers learnt from each other through collaborative EM (i.e. collaborative modelling with an EM approach). For instance, modeller *E* cloned the definition for a SCOUT window from modeller *A*, through the shared interaction history. Although it can be argued that modeller *E* could have figured how out to make such a definition without cloning from A, the collaborative modelling would have been much less effective and productive. Indeed, *E*'s action can be viewed as *learning through example* as a novice systems developer. In the collaborative Sudoku (cf. §6.3), a similar learning situation occurred when modeller *R* asked modeller *S* how to define a set of check rules exhibiting a disjunctive relationship (i.e. logical OR). In the cricket project (cf. §6.4), the learning seems not to be limited to transfer of technical knowledge (e.g. how to use to tool support, syntax of the notations); there is

evidence that the students who knew little about cricket (i.e. domain-specific knowledge) have gradually acquired that knowledge through the collaborative EM process[143].

One may argue that the non-developers (e.g. users, managers, ethnographers) may not wish to acquire development related skills (e.g. making definitions in the EM process). However, as I argued in §3.2, users may invest in such learning if they believe such effort worthwhile (e.g. to increase productivity for future tasks).

Support diverse perspectives and integration

Both the VEL case study (cf. §6.1) and the case study of the cricket project (cf. §6.4) suggest that an EM approach to collaborative modelling is capable of facilitating diverse individual perspectives and integrating them into a shared perspective. In the case study of VEL, two modellers apparently were coming from different trainings (one from electronic engineering, one from computer science). Although the modelling process between them was by and large coordinative rather than collaborative in respect of the degree of engagement (cf. §5.1.1), the practice of EM nevertheless demonstrated how two different yet interdependent agent perspectives integrated into one shared perspective on the "virtual electronic laboratory". In the case study of cricket, the facilitation and integration of diverse perspectives were more apparent. Students modelled the game of cricket with different modes of observation. Such diverse perspectives on the same situation for investigation have their counterpart in groupware development where stakeholders often have different perspectives of how the groupware ought to be, e.g. what feature it should support, how it should behave, what communication structure the groupware should support, etc.

Do not limit the user engagement in development to a sample of users

Although there is provision for *pretend play* (Sun, 1999) by the other human agents, EM does not encourage modelling by *proxy*. This is due to the fact that the referent situation in the real-world cannot be fully apprehended by any person other than the human agent who in that situation. That is to say, only the real users can reflect on and relate to their personal

[143] This potential for EM to integrate learning activities relating to many different subjects and goals was remarked in (Boyatt et al. 2006).

experience. This becomes apparent when the situation involves a large degree of tacit knowledge. However, collaborative modelling with all the stakeholders involved and engaged can be difficult in practice. As mentioned in chapter 3, user participation is impractical in some situations (e.g. there is no user in the product development, the users are too busy in their own activities (cf. Bannon, 1992b)). Furthermore, user participation may sometimes take the form of a token gesture or a political ruse rather than an approach for improving the fitness of the resulting system (cf. §2.5.2). Indeed, user sampling is more or less a socio-technical issue rather than a purely technical issue. Consequently, the potential for an EM approach could be diminished if user sampling takes place in the context of systems development that is highly sensitive to the specific situation, such as is the case in groupware development.

Potential for genuine participation: foundation for human-centred groupware development

Drawing on the above discussion, EM potentially allows active and direct influence by non-developers, supports seamless shifting in participants' roles, facilitates the participants to acquire and to progress their development related skills, and facilitates and integrates diverse perspectives. In the light of the discussion in chapter 3, these in turn support the co-evolution between the participants' understanding and the developing artifact, offering an alternative conceptual framework that takes account of the conflation of and shifting of roles of participants. Indeed, these are the enabling characteristics of genuine participation (cf. figure 7.1 and §3.5). Taking all the above into account, EM for collaborative modelling potentially supports the notion of genuine participation (cf. §2.5.2 and §3.4). Therefore, EM is potentially better suited for human-centred groupware development – since I have already argued earlier in this thesis (cf. §2.5) that genuine participation could be a remedy for human-centred systems development, and groupware development is one special branch of systems development.

7.2.2 Integrating the human and the technological

One of the topical concerns in developing socio-technical systems is how to integrate the technology into the human context so that the use of technology becomes an extension of

the human body and transparent in the human activities (cf. Heidegger 1927; Dourish, 2001). In doing so, tool development has to be human-centred and the tool (i.e. groupware) has to be *grown* in the context of that human activity. As I discussed earlier in this thesis, the essence of HCD can be digested into six principles (cf. §2.5.1). Four of these principles (ii, iv, v, vi) are relevant to the integration of the human and the technological. EM addresses these principles accordingly:

- In EM, modellers continuously negotiate the best configuration (or disposition) through continuous observation, interaction and experimentation (cf. §4.1). These characteristics of EM allow the modellers to examine what is technically possible and to examine whether a particular configuration (and the particular state-as-experienced) is socially desirable. This, in turn, fulfils the HCD principle (iv) and (v).
- In EM, human intervention is possible throughout the process of MwDS (Rungratannaubol 2002). As we seen in the case studies (cf. §6.2 and 6.3), human communication and collaboration took priority over the technological-mediated communication and interaction. This suggests that EM fulfils the HCD principle (ii).
- EM does not distinguish between the context of development and the context of use. This not only facilitates the continuous refinement of the artifact in the context in which the work is carried out, but also avoids the drawbacks that may stem from the separation of the two – e.g. the deskilling of the workers due to formalised procedures of work (i.e. the lack of flexibility in handling exceptions) (cf. Evans and Beynon, 2001). Furthermore, it is possible to alter the actions of a machine agent through human intervention in EM. This all suggests that EM fulfils the HCD principle (vi).

Taking all the above into account, it is appropriate to claim that EM facilitates the integration between the human and the technological. Such a claim contributes to the overall argument that EM supports HCD, which is a prerequisite for supporting the efficacious groupware development perspective.

7.2.3 Enabling flexibility and evolution

As mentioned in chapter 2, HCD needs to cope with the flexible and evolving nature of human activities (cf. §2.4 and §2.5.1). As I discussed in chapter 4, EM artifacts are flexible in structure and always open to changes (cf. §4.1 and §4.3). EM artifacts (or construals) co-evolve in a fashion that is intimately linked with the modellers' personal understanding and experience of the artifact, shaped by the modellers' continuing observation of the referent, and always subject to change (or evolve) in response to the new insights of the modeller. Modelling with definitive scripts (MwDS) in EM neither necessarily follows a predetermined structure nor necessarily follows a predefined path (cf. §4.1.2 and §4.3.5). In fact, EM facilitates evolution at all times because it does not distinguish the development time and the use time (cf. §4.1.2 and §4.3.4). As the analysis of the case studies suggests (cf. chapter 6), the openness, flexibility, and evolving characteristics of EM are preserved when it is practised in a collaborative modelling environment. These characteristics of EM facilitate ad-hoc collaborative modelling (i.e. minimal planning) and *growing instead of building* the artifact in the situation (Brooks, 1987). Taking all these points into consideration, it should be evident that EM facilitates the development of flexible computer technologies that cope with people and allow them to manage and to shape their work, i.e. satisfying the HCD principle (i) in section 2.5.1.

7.2.4 Supporting diverse interaction and communication

When they come to work together, people make use of all the many different interaction and communication channels available to them:

> *"… they [People] take each others' past, present and prospective activities into account in planning and conducting their own work; they gesture, talk, write to each other, and so on, and they mesh these interactional modalities dynamically and seamlessly." (Schmidt and Simone, 1996, p.159)*

HCD not only needs to take the heterogeneous use of interaction and communication into account, but must also address the use of explicit (or rule-based, formal) and implicit (or tacit, informal) knowledge. As Gasson (2003) pointed out, computer technologies that ignore the implicit knowledge counterpart of the explicit knowledge may be unable to play a

significant role in supporting human activities.

EM potentially addresses the use of diverse interaction and communication by facilitating the modellers to reflect on the state-as-experienced at real-time around the developing shared artifact. As the case studies on distributed jugs and collaborative Sudoku reveal, EM modellers interact around and through the artifact in many different modalities – they talked, gestured, leaned towards each other's screen, communicating with and exchanging definitive scripts within the modelling environment.

EM is fundamentally concerned with the use of tacit knowledge in the modelling process. Firstly, the modelling of a referent progresses through practical and tacit knowledge of the referent and the developing artifact. As I mentioned in chapter 4, EM artifacts are not *programs* and are more appropriately regarded as *construals* in the sense of Gooding, which are open for diverse interpretation based on the modeller's experience (cf. §4.1.3). Secondly, EM facilitates modellers who interact and communicate through reflecting on tacit knowledge. For example, in the collaborative Sudoku modelling case study (cf. §6.3), the interaction between modeller *R* and *S* regarding the '*currentstatus*' evidently involved tacit knowledge about the practical use of the observable.

7.3 GroupPIE – a framework for efficacious groupware development

Earlier in this chapter, I have argued that EM is potentially better suited for human-centred development through facilitating *genuine participation*. As I argued in chapter 3, in genuine participation, the roles of participants (developers, users, etc.) become conflated. In fact, the users may demand a high degree of participation as they are deeply engaged in the development process, and contemporary approaches to systems development are not well-suited to this. This motivates us to propose EM as an alternative approach to systems development where human concerns should be given the highest priority, such as groupware development in the context of this thesis. In the previous section (cf. §7.1), I argued that the core activity in groupware development could be viewed as a collaborative modelling process. Viewed from this perspective, it allows us to practise an EM approach to groupware development, as if co-constructing an EM model collaboratively as I discussed in

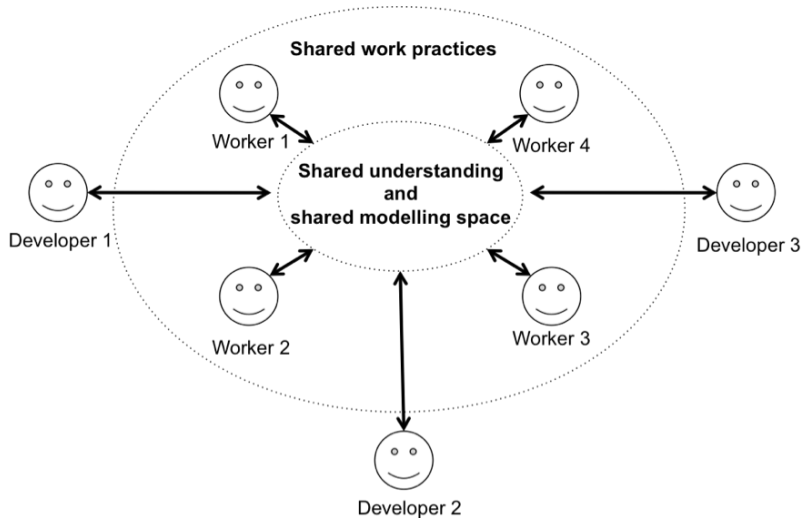
chapter 5 and 6.

Drawing on the conceptual discussion in chapter 3, I define participants as any people who are actively involved in the development of the groupware. Furthermore, I attribute roles to participants according to their actual work in the groupware development process at a particular time. This is to say, participants do not have a dedicated role in our framework. This is to acknowledge and to facilitate the shifting of roles in genuine participation that I discussed in chapter 3. As I already argued in chapter 3, the traditional conceptions of developers and users are loaded with implicit assumptions and are therefore misleading. However, for the purpose of the following discussion, it is helpful to distinguish participants by referring to their main roles respectively. I define *developers* as participants who are systems developers and consultants (e.g. ethnographers, information systems experts), users as workers (i.e. those who mainly carry out work using the groupware), and managers (i.e. those who manage and fund the groupware development).

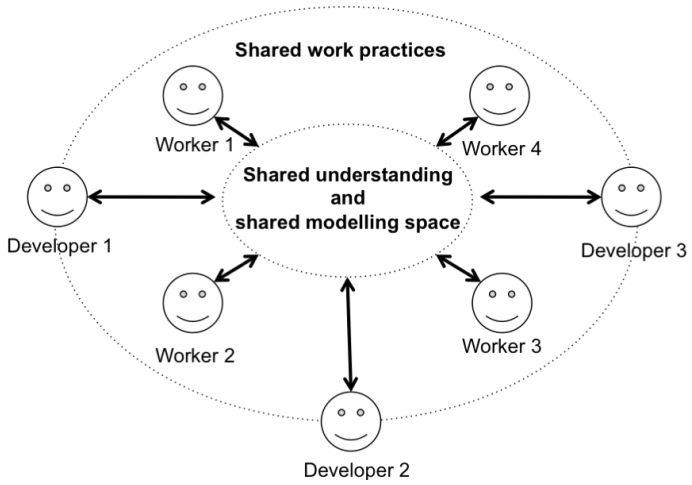
In the following, I present a conceptual framework for practising an EM approach in the context of groupware development. Rather than presenting our approach in a unified process model, I describe it from three perspectives, namely, i) participation, ii) interaction, iii) evolution – hence, GroupPIE. Unless otherwise stated, the groupware development process I describe below follows an EM approach.

The participation perspective

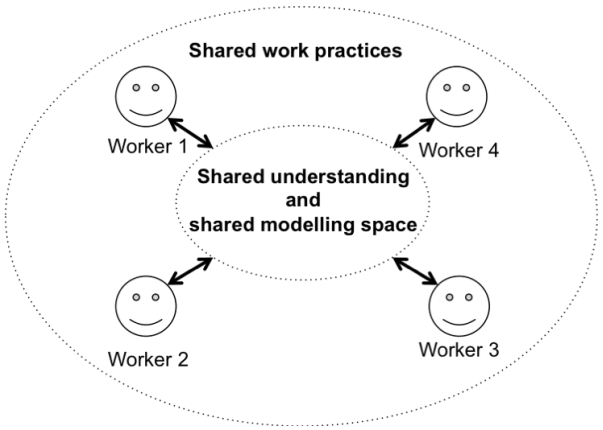
Viewed from the participation perspective, the focus is on the relationship between the developers and the workers. Although this relationship is changing throughout the development process, the changing relationship can be roughly divided into three scenarios. In the initial stage of the development project, only the workers would be familiar with the current work practices and context of work. Although they may interact within a shared modelling space, the developers nevertheless have little or no knowledge about the work practices that surrounding the groupware. In this scenario, irrespective of the purpose of the development and whether it is developer-oriented or user-oriented, the developer is still considered to be the an outsider in the workers' world (cf. figure 7.3a).



(a) Developers contributing from outside of the work practices



(b) Developers observing and working closely with the workers



(c) Developers left the scene and the workers took charge of the system

Figure 7.3 – The participation perspective in the GroupPIE framework

When the groupware development has progressed further, the developers move closer to the workers. The developers may observe closely how the workers carry out the work within the existing work practices. In this scenario, the developers and the workers can jointly develop the groupware and experiment with the developing groupware for the most suitable configuration for carrying out their (future) work practice. With an EM approach, this means that the modellers (i.e. participants of the groupware development) explore and negotiate through and within the model to find an *efficacious* configuration that reflects the state-as-experienced by way of co-constructing and co-evolving a shared EM model through the EM process (in the sense of co-evolution as it was described in chapter 5, and studied through the case studies in chapter 6). The motive behind this collaborative modelling with EM is to facilitate *genuine participation*. The virtue of collaborating and experimenting within a shared modelling space and workspace is that the participants (or modellers) can reflect upon their immediate experience when interacting with developing groupware (i.e. the EM artifact) in the shared modelling space (i.e. the EM environment). As in PD, developers and workers would mutually learn each other's practices through a high level of participation and engagement within this collaborative modelling process. However, as figure 7.3b depicts, the developers can never entirely enter into the workers' world, they can merely observe and collaborate with the workers at the boundary of the workers' shared work practices.

The motive behind such a high level of participation and facilitation of engagement is to shift the ownership of the groupware and the development process to the workers. This is based on the belief that workers are the owners of the problem (i.e. the activity which the groupware is to support), of the groupware, and of the groupware development process. In keeping with the idea of *efficacious groupware development* (cf. §2.5), the development process for groupware should encompass its lifespan from *conception of the groupware* to *disposal of the groupware*. This implies that the workers have to take charge of the *groupware development* at some stage (cf. figure 7.3c). Where 'customisation' in groupware development has been conceptualised in the past (e.g. design-in-use (Henderson and Kyng, 1992), tailoring (Wulf, 1999), evolution (cf. §2.4.2), etc.), it has been rare for the groupware developers to stay within the customisation process (whatever it is called) in the traditional conception of systems development. For this reason, I argue for *genuine participation*

through collaborative modelling with EM in the GroupPIE framework. As argued in section 7.1, EM is potentially better suited for *genuine participation*, which allows the shifting in ownership and the shifting of roles (i.e. from users to developers in the sense of the traditional conception) to take place seamlessly.

The interaction perspective

Viewed from the interaction perspective, the focus is on facilitating diverse modes of interaction among the participants (or modellers) within the groupware development process (or collaborative modelling process). In chapter 5, I discussed a number of modes of interaction that may take place during a collaborative modelling process and I also argued that the modes of interaction among the modellers might evolve from time to time, due to different needs in the current activity or task at hand (cf. §5.1.3 and §5.1.4).

As in collaborative modelling, it is impossible to prescribe a framework that covers all the potential patterns of interaction that may take place in the context of efficacious groupware development due to the dynamic, situated, and complex nature of human interaction. One way to conceptualise such diverse modes of interaction in the context of groupware development, is to recognise it with a *cloud*[144]. The notion of cloud here is similar to what McLoughlin et al. (2009) refer to as the "interaction cloud" in a recent workshop in ECSCW 2009. McLoughlin et al. (2009) explains that the notion of interaction cloud reflects "the complex patterns of interaction, sharing and exchange that unfold around assemblies of mobile and embedded technologies", in which "technology is emerging that allows people to shift between multiple contexts (mobile/embedded), facilitated by a connection to remote computing resources (clouds)".

As figure 7.4 depicts, interaction among the participants within the development of groupware should be *reconfigurable* according to the situation. With an EM approach, the participants of the groupware development process may interact, observe, and experiment in diverse modes. In the light of the participation perspective (as discussed above), the

[144] The cloud metaphor is borrowed from cloud computing (Weiss 2007), which has recently emerged in the IT industry.

workers will interact, observe, and experiment with the developing artifact during the early stage of development (viz. figure 7.3a to figure 7.3b) as if the developers are transparent in the process. The workers will continue to interact in the same way, and observe and experiment in a similar way when the developers withdraw from the process.

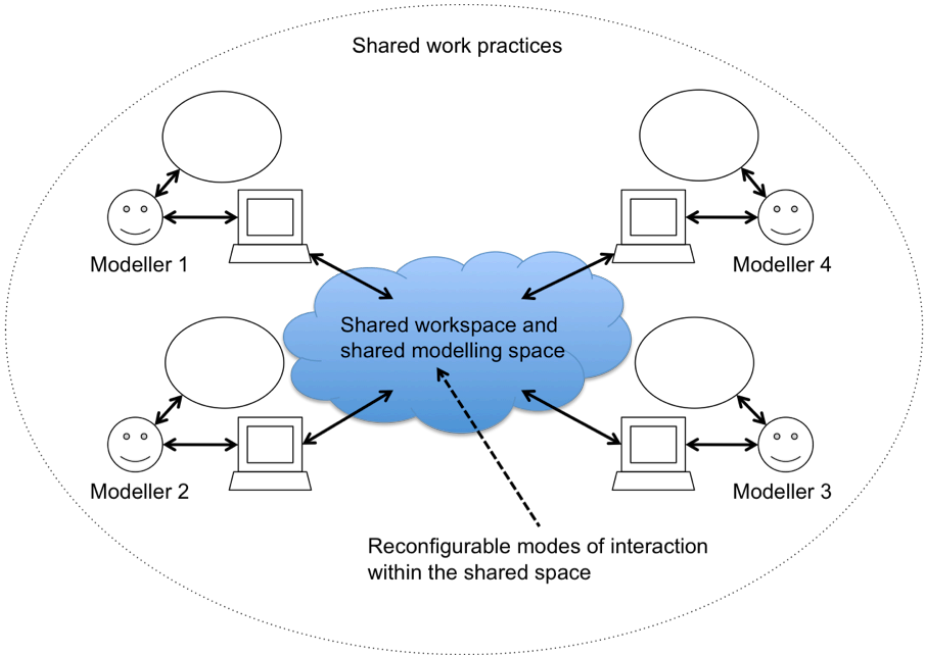


Figure 7.4 – The interaction perspective in the GroupPIE framework

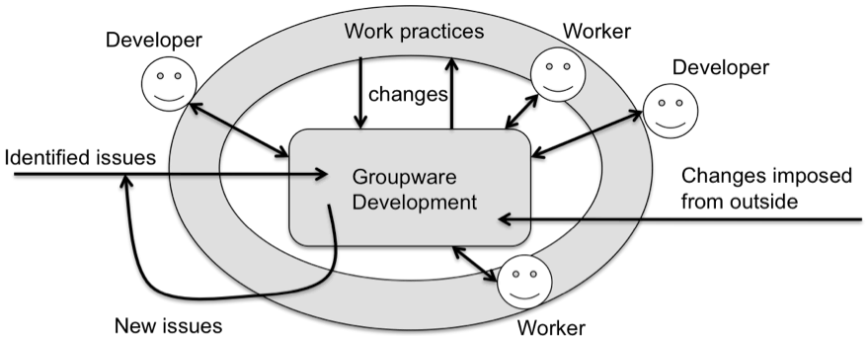


Figure 7.5 – The evolution perspective in the GroupPIE framework[145]

[145] This diagram takes its inspiration from the SPORE process in (Sun 1999, p.233)

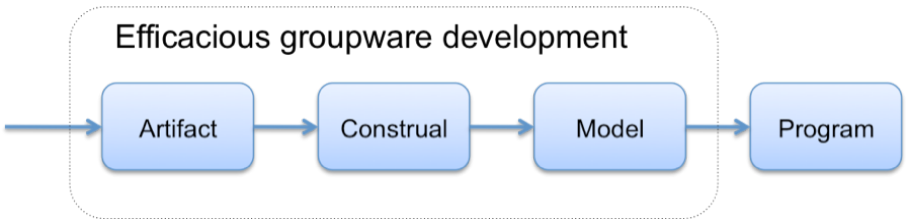


Figure 7.6 – The scope of efficacious groupware development in relation to ACMP

The evolution perspective

Viewed from the evolution perspective, developing groupware with the GroupPIE framework can be thought of as *growing* groupware *in situ*. That is, the groupware is co-evolved with the group of workers that eventually takes charge of it (and the developers) within the environment which the work is taking place. Figure 7.5 – though it might be overly simplified from a real world groupware development situation – illustrates the interplay between the key entities (e.g. participants, factors, etc.) around and surrounding the evolutionary process in the GroupPIE framework.

In chapter 4, I described how an EM artifact can be viewed as a construal, a model, or a program when the observables, dependencies, agents, and agencies (ODA) achieve a certain degree of stability. In the context of efficacious groupware development, however, the EM construal may never be regarded as a program. This is due to the fact that the context for the groupware operation (viz. context of the work, the work practices, the group itself, the individuals within the group, etc.) is ever-changing and evolving at all times. Subsequently, ODA in this context may never be considered as finalised in the sense of traditional programming (Beynon et al., 2006). Viewed from the evolution perspective of the GroupPIE framework, the groupware is always *growing* like an organic process with an extended lifetime. Figure 7.6 depicts the scope for efficacious groupware development from an EM perspective.

7.4 Limitations for the GroupPIE framework

The formulation of the GroupPIE framework is based on previous research into EM and the research into EM for collaborative modelling that I discussed in chapter 5 and 6. Although I have argued the potential of EM in the collaborative modelling context and have shown the

potential of EM in the groupware development context (when practising in the GroupPIE framework), there are at least three limitations for the GroupPIE framework:

1. Firstly, I have only considered collaborative modelling and groupware development with EM in small groups which have no more than a dozen of participants (or modellers). It is intended to be an approach for small teams in small and medium sized enterprises (SMEs) to *grow* groupware. The case studies that I discussed in chapter 6 were all aimed at this context. Certainly, larger teams may face a similar socio-technical challenge when developing software systems support. However, the scale of the case studies cannot indicate whether an EM approach, with the GroupPIE framework, can be practised in larger teams. Furthermore, it can be argued that software systems for a larger population of 'users' are not groupware but multi-user enterprise systems.

2. Secondly, the GroupPIE framework has not taken safety critical properties of the developing system into account. In the safety critical domains (e.g. healthcare, space mission, avionics, automobile, etc.), priority in the development is given to ensuring that the resulting system is safe, secure, and reliable (cf. Sommerville, 2004), rather than to the social aspects surrounding the development process. Failure of the system can be disastrous and often costs human lives. It would be risky to adopt the GroupPIE framework for groupware development in safety critical domains.

3. Thirdly, the GroupPIE framework merely affords the potential for the participants of the groupware development process to grow the groupware and to negotiate for an *efficacious disposition* from time to time (cf. figure 7.5), it does not explicitly focus on the existing political issues in the group. It is possible that the research described in this thesis could be extended to study the political impact of the GroupPIE framework, but such research is nevertheless out of the scope of this thesis.

7.5 Towards efficacious groupware development: vision for GroupPIE

Earlier in this thesis (cf. chapter 2), I argued for an *efficacious disposition* perspective for groupware development (i.e. efficacious groupware development), which views the previously separated development, use and evolution processes as *a single, holistic, organic, and situated process* that continuously arranges existing ingredients in the situation to find a suitable configuration in the evolving context throughout its lifespan. This perspective takes account of the fact that the design and development of groupware goes over the preconceived boundary between the context of development and use. Rather than drawing a boundary between design, development, and use, this perspective conflates all preconceived contexts. Subsequently, the roles of participants are also conflated (cf. chapter 3). Due to the holistic view of contexts and participant roles, efficacious groupware development is different from other perspectives on groupware development and evolution, e.g. meta-design (Fischer and Scharff, 2000). Furthermore, such a "lifelong" perspective is controversial in the context of traditional systems development, but not in the light of lifelong human activities, e.g. lifelong learning (cf. Beynon and Harfield, 2007).

In this chapter, I have proposed a plausible agenda for practising an EM approach to efficacious groupware development. This agenda consists of two parts: i) orienting groupware development as collaborative modelling (cf. §7.1); ii) a framework (viz. GroupPIE) that sketches how EM might be practised collaboratively in the context of groupware development as it were in a collaborative modelling context (cf. §7.3). Central to the GroupPIE framework is the notion of *genuine participation*. Earlier in this chapter, I argued that EM is potentially better-suited for human-centred development and for efficacious groupware development, because it potentially fulfils seven recommendations in facilitating the notion of genuine participation (cf. chapter 3). In genuine participation, participants are treated as human beings who are acting with their knowledge rather than persons who have preconceived roles as in traditional systems development. Hence, the context for participation and action can be thought as a single, shared, situated, and conflated modelling space. As argued in chapter 2, the issues in the socio-technical

challenge are intertwined so that they cannot easily be tackled independently. Insofar as EM supports genuine participation, this allows the issues to be considered and to be addressed as a whole, rather than independently. What is more, these issues can be addressed when they arise throughout the EM process. To some extent, collaborative modelling through an EM approach is similar to collaborative prototyping (e.g. Stavash and Chadh, 2000). However, an EM approach to systems development is different from prototyping in at least three ways. Firstly, a conventional prototype may be thrown away, but this is not the case in the EM approach that I am proposing for efficacious groupware development. Secondly, prototyping maintains a developer-user conceptual gap: prototypes are constructed by the developers and then sent to the users for evaluation. Users then give feedback to the developers. Sun (1999) refers to this process as *backward feedback*. Thirdly, EM allows the modellers to interact with and through the EM model in an open and flexible manner (cf. chapter 4).

Since EM is a human-centred approach to computer-based modelling, it does not follow a prescription paradigm for systems development (Beynon et al., 2006, Wong, 2003, Sun, 1999). As Harfield (2008) writes

> *"… one of EM's powers is that within a model 'everything is up for grabs' at any time during construction and use [i.e. the EM process – EM does not distinguish between construction and use]" (ibid, p.223)*

The direction for developing the artifact (i.e. the groupware) is determined by the modellers' past and immediate experience of continuous interaction with and observation of the referent (i.e. the mental model of the groupware in the modellers' mind) and of the developing artifact (i.e. the actual groupware evolving in the modelling environment), which cannot be predicted and framed in advance. Consequently, the GroupPIE framework sketches three perspectives (viz. participation, interaction, and evolution) – which cover different levels of activities – in the context of efficacious groupware development when practising an EM approach, rather than explicit and step-by-step guidelines for practising EM.

The GroupPIE framework is based on the potential of practising an EM approach to collaborative modelling. In this framework, the participants co-evolve the EM model at different degrees of engagement, different participation schemes among the modellers, and different modes of interaction through a shared modelling space, as if they were practising an EM approach to collaborative modelling (cf. chapter 5). The case studies in chapter 6 strengthen this argument as they have shown that EM models can be co-constructed and co-evolved in different collaborative modelling contexts.

Efficacious groupware development is a continuous disposition of the groupware (through arranging the agents internal and external to the groupware) within the evolving human context that includes the work practices and the group activities, which surround the groupware and are to be supported by the groupware, throughout its lifespan. With this in mind, the GroupPIE framework facilitates *efficacious disposition* in the following respects:

> *Initial efficacious disposition* – the developers (i.e. participants including ethnographers and experts other than the workers) and the workers co-determine whether the development should begin from scratch (i.e. no definitions in the modelling environment prior to the development) or be based on an existing model. As I demonstrated in the collaborative Sudoku case study in section 6.3, it is possible to reuse an existing EM model to kick-start the collaborative modelling[146]. On the one hand, it is more efficient to begin the development from an existing EM model. On the other hand, reusing an EM model can be risky because there will no guarantee the model would capture the experience of the modeller as the links between observables and their contexts of observation may be broken[147] – hence, the benefit of practising an EM approach may be partially lost. Since the initial stage in the GroupPIE framework is a selection between two possibilities, it can then be viewed as *finding an efficacious disposition (or a condition) for the groupware to grow*.

[146] The EM model used for kick starting can be thought as a 'seed' EM model (Sun, 1999; Chen, 2001).

[147] If the modeller is not the *owner* of the reusing model, the model can merely be regarded as a traditional program rather than an EM construal.

Participative efficacious disposition **–** based on the initial model, the participants co-evolve the EM model. The participants continuously observe, interact, and experiment with the evolving artifact collaboratively at different degrees of engagement, different participation schemes among the modellers, and different modes of interaction in a shared modelling space. Since the participants are improving the configuration continuously, it can be viewed as *seeking an efficacious disposition so that the groupware and the group activities will work together harmoniously*.

Evolutionary efficacious disposition – as the developers withdraw from the efficacious groupware development process, the workers take over the process and continue to co-evolve the groupware (i.e. making redefinitions to the EM artifact) according to their needs and in response to the evolving work practices in the rest of the groupware's lifespan. This can be viewed as *a continuous seeking for an efficacious disposition so that the groupware and the group activities continue to work harmoniously*.

Despite the potential that the GroupPIE framework may entail, realising efficacious groupware development in practice is still challenging because of the following issues:

Lack of good EM tool support for collaboration – the tool support for DEM (viz. dtkeden) and its variants that I used to carry out the collaborative modelling case studies in relation to this thesis (viz. dtkeden-blackboard, dtkeden-cm) were still in an experimental stage. None of this tool support allows dynamic switching of the interaction modes faultlessly. Furthermore, as we observed in the collaborative modelling case studies, the modelling was disrupted a couple of times due to the instability of the tool. Apart from the lack of collaboration support[148], the modellers were often distracted by the error messages caused by mistaken definitions input in the shared modelling space.

[148] However, the tool support for collaboration could be added on-the-fly in the modelling process as needed. As one of the collaborative modelling case studies demonstrated, the modellers can develop and evolve such support in a situated manner.

Motivation for participants – GroupPIE may suffer similar issues to other participatory approaches for socio-technical systems[149] development that involve a high-degree of user participation, e.g. PD (Schuler and Namioka, 1993), meta-design (Fischer and Scharff, 2000), informed participation (Brown et al., 1994, Fischer and Ostwald, 2002). For instance, Fischer and Scharff (2000) pointed out that stakeholders (i.e. workers, managers, etc.) must be motivated and rewarded for the extra work that they may incur through their participation. This echoes Grudin's (1993) observation in groupware development. A similar motivation-reward issue is also found in relation to the acquisition of development-related skills. Blackwell (2002; 2004) argued that the users (or workers) would invest resource to acquire such skill if they believe that the potential benefit of that investment is greater than the outcome of not investing. A common issue in these participatory approaches is that they are more or less developer-oriented – a problem that I highlighted in section 2.5.1. In contrast, GroupPIE is a hybrid approach that combines developer-oriented and user-oriented development. It is a framework specially designed for practising an EM approach in the context of groupware development. The ownership of the process is gradually shifted from the developer to the workers through *genuine participation* (cf. §7.3). By shifting the ownership to the workers, the responsibility is also shifted to the workers. In this way, the workers will have to take care of their own groupware, just as a house owner has full ownership and responsibility for taking care of their own house when she gets the key from the house builder[150]. Nevertheless, it is not too clear how far the GroupPIE can go in relation to the lifelong aspect of efficacious groupware development.

[149] Groupware is an instance of a socio-technical system, but the size, the scope, and the users of groupware are on a smaller scale. For instance, Facebook (http://www.facebook.com) can be classified as a socio-technical system but not as a groupware because the size of the groups on Facebook is usually more than a hundred.

[150] This analogy is only valid for the ownership and responsibility handover. It is not perfect because the house builder is not likely to transfer their building skills to the house owner.

Groupware development in the economic culture – in my own industrial experience[151], most systems development projects (whether socio-technical or not) prioritise cost and time over quality (in terms of the factors surrounding the project triangle (cf. Lewis, 2005)) due to the existing economic culture. As argued earlier in this thesis, it is unreasonable to facilitate a reduced degree of participation and to expect the benefit of a high degree of participation (cf. chapter 2 and chapter 3). If participatory approaches do not cope with the demands of such a cruel economic culture, their promise, future, and success in groupware development (and in generic systems development) could be diminished. It is not clear, at this stage, whether radical participatory approaches, such as the GroupPIE framework (cf. §7.3), will flourish in the *cultures of participation* (Fischer, 2008; 2009).

Despite the issues mentioned above, it should be evident that EM is promising for efficacious groupware development. The study of EM for collaborative modelling (cf. chapter 5) and the GroupPIE framework nevertheless has set out a plausible direction for further development for collaborative EM and for realising efficacious groupware development. With the GroupPIE framework, the possibility of practising an EM approach to efficacious groupware development is taken one step forward.

[151] Prior to my postgraduate studies, I worked in the IT industry as a systems developer and a systems administrator between 2000 and 2003.