Collaborative Workflow Editing for Online Collective Action

by

Sébastien Molines, DUT

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

September 2010

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Sébastien Molines

September 13, 2010

To Oonagh, and our baby Théo

Acknowledgments

I am deeply indebted to Dr David Lewis, my supervisor, for his support and patience throughout the course of my work on this dissertation, and for sharing his knowledge and vision in a manner which kept me from going astray while allowing me the freedom to make my own choices.

I also wish to express my profound gratitude to Dominic Jones, who kindly chaperoned me in the planning stages of the evaluation, assisted me in its conduct, and used his influence to secure participants of the highest calibre for the user study.

Sébastien Molines

University of Dublin, Trinity College
September 2010

Collaborative Workflow Editing for Online Collective Action

Sébastien Molines, M.Sc.
University of Dublin, Trinity College, 2010

Supervisor: David Lewis

Workflows and business process modelling have been constrained almost exclusively to business settings, where they are entangled with notions of control and hierarchy. In this dissertation, we aim to free workflows from their acquired notion of power and conceive of them as the documented models of collaboration which they actually represent. Advances in online collaboration have spun powerful new forms of collective action, which have reached a level of sophistication not unlike the complex business interactions found in industry. In this work we apply to workflow modelling the same factors which have allowed successful forms of online collective action to flourish, build a prototype and carry out group trials. In establishing how to collectively model workflows in the absence of central authority, workflows may be applied to new contexts and contribute to the continued growth of new modes of value creation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Workflow design, the act of mapping out interactions between participants in a distributed activity, is generally a top-down undertaking: managers conceive business process workflows and subordinates apply them. The presence of management authority, whose essential function is to control business activity, tends by nature to facilitate the adoption of workflows. This has led to the development of commercial workflow management solutions adapted to centralized management scenarios.
However workflows can provide value irrespective of the existence of central authority; many of their benefits can be as useful to self-managed groups as they are to tightly-structured enterprises. Indeed, online collective action requires a sufficient level of organization, which can be supported by the adoption of a workflow. But while Web technologies have done much to facilitate collective action by groups of individuals, collective workflow management technology adapted to the specifics of online collective action is still lacking.

1.2 Motivation

The Web 2.0 paradigm, also known as the read-write web[1][2], has accompanied and facilitated an ongoing shift in industry to more distributed, less centralized decision-making, which is advantageous in terms of business agility and receptivity to customer needs[3]. Beyond industry, the same advances in web collaboration have also enabled new forms of collective action, supporting emerging modes of value creation that distil the “wisdom of crowds”[4]. Our aim, therefore, is to establish whether the same principles can be successfully applied to workflow modelling, and to propose a collective approach to workflow design.

1.3 Hypothesis

This dissertation seeks to establish that appropriately designed Computer Mediated Communication technology can enable groups of individuals with no central authority to successfully participate in the collective conception and refinement of workflows through consensus-building mechanisms.

1.4 Research Approach

1. We examine the key sociability, usability, design and technological issues which influence the suitability of a workflow system adapted to this purpose. This allows us to derive corresponding requirements.
2. We build a prototype system that fulfills these requirements and allows continuous refinement and adaptation, capabilities that are problematic[5] in rigid top-down workflows.
3. We evaluate the prototype's suitability by conducting a trial experiment, then draw conclusions and propose further work.

1.5 Outline

We begin this dissertation with an analysis of the relevant research topics, ranging from online collaboration to workflows, in chapter 2.

In chapter 3, we gather a set of functional requirements, which then inform our design choices and allow us to identify the key concepts of a proposed solution.

In chapter 4, we discuss the specifics of a prototype we implemented and describe its key features.

In chapter 5, we describe how the prototype was evaluated and we analyse the data collected to draw conclusions from it.

In chapter 6, we suggest what further research can be conducted from our findings and how this work can be extended.

Finally, we end this dissertation by drawing our conclusions in chapter 7.

Chapter 2

State of the art

2.1 Online Collective Action

The Web, and the ease with which it allows people to interact, has facilitated new forms of group participation, leading to a number of notable successes such as open-source software and Wikipedia. These successes have made Wikipedia and open-source software communities the subject of much research. One distinction to be made, however, is that Wikipedia and open-source software are the result of collaborative production, while collective action is "the hardest kind of group effort, as it requires a group of people to commit themselves to undertaking a particular effort together, and to do so in a way that makes the decision of the group binding on individual members"[6]. Thus collective action implies the existence of group governance.

2.1.1 Self-Governance

The study of the emergence of collective action governance pre-dates Web 2.0[7]. The Tragedy of the commons[8] is frequently used to exemplify the issues that arise from the conflict between self-interest and collective interest. These issues tend to lead to the establishment of rules and governance.

Research on long-established, successful commons-based communities can be applied to identify principles which can be transferred to online communities, as Viégas, Wattenberg and McKeon have observed in their analysis of Wikipedia[9]. Similar rules apply, as organizing any type of collective action – be it online or offline – requires addressing the problems of "creation of institutions, monitoring mechanisms, arbitration, and conflict resolution"[9].

Self-governance is evolutionary; socio-technical systems gradually evolve more formal controls as they mature[10]. Adequate management features are therefore particularly difficult to implement, and require a level of flexibility that tends to be lacking in the traditional management systems, which are often rooted in the traditional enterprise perspective of centrally-managed access control. Research into progressive self-management has led to the development of an analytical framework for the management of online communities[11]. Presently, however, Wikipedia lacks pro-active functionality to prevent rule breaking and relies on the intervention of the community to enforce rules and apply corrective action, with remarkable effectiveness[12].

Wikipedia is a very informative example of online governance. Due to its exposure, Wikipedia is an attractive target for lobbyists, political or governmental organizations and people concerned with promoting their self-interests, who may be liable to abuse the policies of Wikipedia such as the neutral point of view principle[13] in order to advance their personal interests (some of whom have been publicly exposed for doing so[14]). This is mitigated by the traceability of contributions and the openness of the data which, combined with the large number of observers in the community, provides a line of defense against this type of abuse. In our case, we expect workflow modelers to be part of a much more cohesive and much smaller scale community than in Wikipedia, and issues of conflict of interest are less likely to be a concern. More generic issues of malicious activity are examined in the security section in Appendix E

2.1.1.1 Consensus Building

Wikipedia uses *talk pages* to facilitate deliberations and conflict resolution[15]. These pages are precious artifacts for the study of online collaboration.

Wikipedia includes complex processes for reaching consensus, in particular for the purpose of electing administrators or making editorial decisions[16]. For example, the process of deleting articles of poor quality involves a five-day period of discussion followed by vote-taking, but the decision is not automatically dependent on the outcome of the votes—it is the administrator who makes a final judgement based on the deliberations and these votes[16].

Interestingly, research into argumentation support technology has pointed out that the progressive evolution of the types and goals of argumentation tends to be insufficiently supported by overly rigid software systems, causing fluctuating mismatches between what the technology provides and what the community needs[17]. In the case of workflows, however, the deliberation needs can reasonably be expected to remain relatively static and constrained, although an empirical evaluation of the deliberation support is required to support this assertion.

2.1.2 Online Communities

There is no universally agreed definition of what an online community is. Jenny Preece, in her seminal book on the topic, *Online Communities: Designing Usability and Supporting Sociability*, defines it by its constituent components[18]:

1. People
2. A purpose
3. Policies
4. A computer system

Online communities are varied and their sociability is difficult to analyze[19]. Research into online communities is relatively new and complex, and borrows "from sociology, communications studies, computer-supported cooperative work, and social psychology"[18]. It has led to specific recommendations to help Web developers design their applications adequately in order to support the establishment of a healthy, stable online community[18]. To this end, a number of factors have been identified:

2.1.2.1 A Shared Purpose

A clear, shared purpose is what draws people to a community. When clearly stated, the community purpose has a stabilizing effect that reduces hostility, as it ensures that participants have appropriate expectations and common understanding and "helps to deter casual visitors who lack commitment"[18].

2.1.2.2 Sociability and Usability

Sociability drives usage: Web applications that have the most success in capturing committed users and nurturing participation generally are those which provide the best sociability and usability support[18]. Online social presence is also an important aspect; visible presence ("online now") helps sustain online communities and contributes to the reputation of its members[20].

2.1.2.3 Rich Communication Support

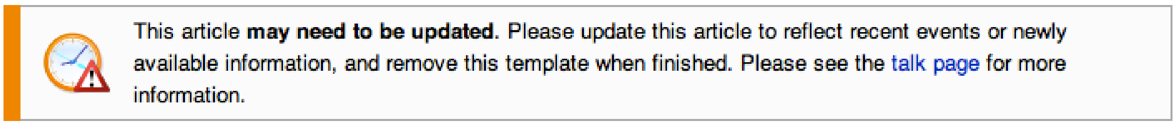
Support for interaction and communication is key to allow a thriving community to grow. The ability to convey empathy is important, but is particularly difficult to achieve online as it is overwhelmingly conveyed in nonverbal forms[21]. Emoticons are often used in an attempt to reduce misinterpretation of textual communication, with some level of success[22], and acronyms such as 'LOL' are also commonly used to overcome the loss of nonverbal cues[18].

2.1.2.4 Visibility

Visibility is important, and care must be taken "to give users a sense of who else is active in the community and what they are doing"[18]. This is exemplified by Wikipedia, which provides a full history of all the contributions of its users, including their participations in online deliberations on the *talk pages.*

Additionally, Wikipedia provides visibility on work in progress by displaying 'templates' (frames with visual markers inserted in article pages)[9] such as "this article has multiple issues", "this article is a stub" or "this article may need to be updated", shown on figure 2.1)

A further example of a visibility feature can be seen in Oryx[23], which provides an innovative way of communicating the maturity of a diagram through the use of the sketchy visual style[24], as illustrated in figure 2.2. The hand-drawn stylistic ef-



This article **may need to be updated**. Please update this article to reflect recent events or newly available information, and remove this template when finished. Please see the talk page for more information.

Figure 2.1: A standard Wikipedia template reporting a content issue

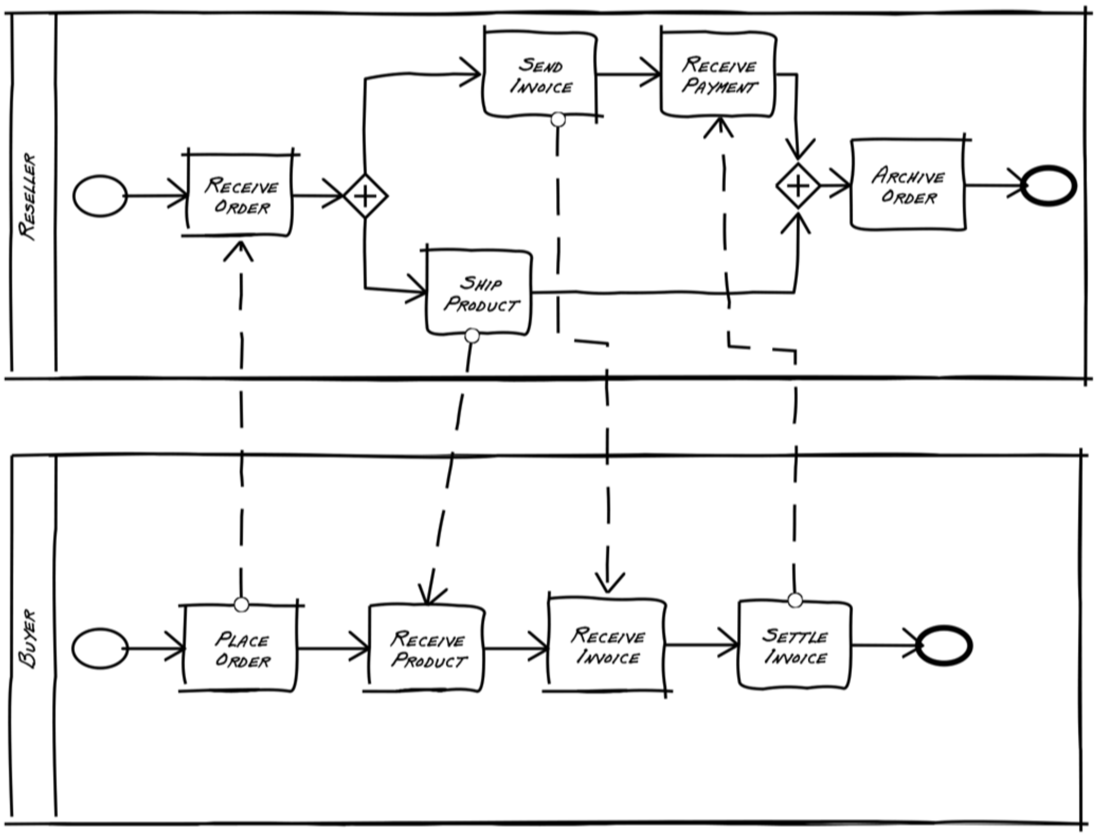


Figure 2.2: A model exported in sketchy style in Oryx

Source: Oryx Developer Network[24]

fect intuitively convey the message that the document is intended to be treated as a draft. The authors of Oryx expect that this visual message has the effect of lowering the reluctance of potential contributors to propose amendments and thus stimulates more contributions than if the document were presented in its final, publishable form. This feature is limited in its implementation, however: it is only accessible as a file export setting and thus cannot be seen at edit time. It would be interesting to allow such a visual style at design time so that collaborators who are themselves using the application (rather than receiving a model as exported file) are encouraged to treat the model as a draft.

Another visibility-oriented metadata example is Wikipedia's feature for highlighting minor edits. Wikipedia's help article on *Minor Edit*[25] summarizes it as follows:

> "Checking the minor edit box signifies that the current and previous versions differ only superficially (typographical corrections, etc.), in a way that no editor would be expected to regard as disputable.
> Any edit that changes the meaning of an article is not a minor edit, even if the edit concerns a single word, and it is improper to mark such an edit as minor."

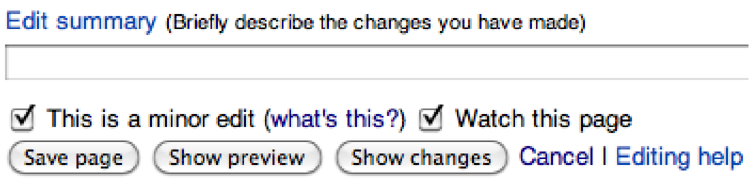

Edit summary (Briefly describe the changes you have made)
✓ This is a minor edit (what's this?) ✓ Watch this page
Save page
Show preview
Show changes
Cancel | Editing help

Figure 2.3: Marking an edit as minor in Wikipedia

Edits marked as minor at submit time are indicated with the underlined letter '<u>m</u>' on the history page. Users are expected to mark their edits as minor where appropriate, using checkbox "This is a minor edit" on the Wikipedia page shown in figure 2.3 prior to submitting changes. Given the fact that users are relied upon to carry out this marking, there is considerable scope for omissions and misuse, which compromises the reliability of this metadata.

Effective change tracking plays an important part in supporting visibility in collaborative systems. Ellkvist [26] proposes a system to manage concurrent changes in scientific workflows – but with a particular focus on real-time editing, which is not our priority scenario.
For Wikipedia, research has led to the development of visualization models such as "History Flow"[27] that could be transposable to workflows. However, differences between graphical editing and article editing in qualitative and quantitative terms (as usage contexts are quite distinct) need to be factored into the design of change visualization.
Oryx[23]-based BPMNCommunity.org[28] includes basic support for change visualization, based on filling elements that have changed with a colour representing the change made (e.g. red for added, green for edited).

2.1.2.5 Accessibility

While accessibility per se is not a requirement for the establishment of a thriving online community, there is a particularly strong moral case for supporting accessibility and inclusion in general, as online communities can be especially beneficial for people with disabilities[18]. Workflow editing tends to rely heavily on graphical representations models such as BPMN, and thus tends to assume that all contributors are sighted people. Alternative representation schemes allowing the description and amendment of workflows in a non-visual manner are required if workflow editing is to be inclusive. Virtual communities can be open to the participation of visually impaired users through the provision of concise textual representations expressed in Braille or synthesized speech[29]. While this is theoretically possible given the human-readable XML format of BPEL, it remains a usability concern in practice, and the issue of inclusion in our context is a difficult problem calling for further research.

2.2 Workflows

2.2.1 Introduction

Business process modelling originates from the turn of the 20th century, when it was applied to optimize industrial tasks as per the Taylorist method of scientific management[30].

With its productivity and efficiency improvements came social degradation[31], as Taylorism and the business process reengineering movement that succeeded it tended to down skill and dehumanize employees while ignoring their needs and motivations[32]. Yet it eventually spread to most of the manufacturing sector and beyond, also shaping—albeit with less dramatic effect—some of today's service sector, a prime example being call centers[33].

The negative impact of business modelling arises from the issue of control and from its historical top-down, coercive nature. However, when business processes are reengineered in an inclusive manner that results in an increase of worker discretion and autonomy, which sadly is not generally the case[34], job satisfaction and occupational health improve[34]. Hence best practices recommend allowing the people directly involved in the operation of business processes to influence their design, partly because doing so has the effect of reducing resistance[35].

Additionally, top-down business modelling is restricted by the "differences in perception of those who 'design', in contrast to those who 'use' technology"[36]. This is a strong advantage of direct participation, which can result in finer-tuned business process models. Indeed, case studies have concluded that "the decisions made by self-managed work teams are extremely effective because those making the decisions – the team members – are the most knowledgeable persons about the work"[37].

Workflows are essential for documenting business processes. They specify the participants, their interactions, the information that is exchanged between them and its control flow. They facilitate analysis, helping to identify critical paths, repetitions and bottlenecks, and thus help to optimize and refine business processes. In practice, workflows provide most value and are most often used to document repeatable tasks, or routine aspects of work activities[38].

2.2.2 Workflow Typology

In my review of the literature on workflow languages, I have identified the following attributes that will influence the scope of the work and the experimentation and evaluation criteria:

2.2.2.1 Notational Versus Semantic

1. Some standards relate exclusively to models of graphical notation. This is the case of BPMN (Business Process Modeling Notation), a workflow graphical representation standard, and of a number of other proprietary notations. UML Activity Diagrams[39], which are sometimes used to represent workflows[40], also fall into the notation category.

2. BPEL, on the contrary, is an execution model that has no specific graphical representation model. It can therefore be categorized, together with other executable workflow languages, as semantic rather than notational.

3. Some workflow standards are concerned both with notation and semantics. This is the case of XPDL, the Workflow Management Coalition (WfMC)[41] format most commonly used to serialize BPMN diagrams and other notation diagrams. XPDL includes precise information about the visual aspects of the diagram – there is a myriad of ways to lay out one same workflow diagram – which are important for human needs (given our visual recognition based on photographic memory and our cultural conventions such as left-to-right ordering)

Mappings between notational languages and the BPEL executable language have been proposed[42], however translation is not a straightforward task. In particular, research has shown that translating from BPMN to BPEL remains problematic[43], an unfortunate situation given that they are the dominant standards in their respective categories (albeit maintained by distinct standards bodies, the OMG and OASIS).

2.2.2.2 Expressiveness

Workflow standards all have different expressive power[44], i.e. limitations in what can and cannot be represented. Research on workflow patterns, mentioned in the next paragraphs, has led to the standardization of a variety of patterns which may or may not find corresponding constructs in a given workflow language. These patterns help evaluate the expressiveness of existing workflow standards; in particular, BPMN[45] and WS-BPEL (and extension of BPEL designed for Web Service orchestration)[46] have been methodically evaluated to uncover their capabilities and shortcomings. Pattern research has also lead to the design of the highly expressive YAWL language[47].

2.2.2.3 Conceptual Versus Executable

Intents differ in the design of workflows:

1. Business analysts and persons concerned with mapping out interactions and data flows may do so with the aim of better understanding, documenting and analysing business processes. In such cases, they "rely on loose and generic modelling formalisms which cannot provide any basis for experimentation and quantitative evaluation"[48]. In other words, the workflow is abstract and is not, and need not be, executable.

2. As part of business process management system[49], workflows are designed to drive the automatic orchestration of business processes. In such cases, completion and correctness of the data types of the messages being passed between processes are required.

3. In Service-Oriented Architectures, workflows are used for the composition (or orchestration) of Web Services into a cohesive solution, called a composite application[50]. The OASIS standard WS-BPEL[51] (also called BPEL4WS) appears to have become the de-facto standard, although many others exist[52].
   Such workflows are executed by workflow engines – common open source workflow engines include Apache ODE, jBPM, Open Business Engine.
   Executable workflows also include scientific workflows[53], which are the subject of much recent research and relate to the specific domain of scientific computation, often involving grid computing.
   Human tasks can be included in BPEL orchestrations: BPEL4People[54] is an extension to WS-BPEL that adds support for tasks to be carried out by human actors. Microsoft also provides similar support in Windows Workflow Foundation, and defines a human workflows as supporting "both human and system interaction according to business rules"[55]. Human workflows are part of a system that manages and tracks tasks for the attention of human actors, for example alerting them to the need to carry out a specific task as soon as the information

required to begin this task becomes available.

The scenario of Web Service composition is quite distinct from our problem area and will not be considered. Our focus is on business process modelling rather than the workflow-driven integration of software systems.

An interesting side effect of executable workflows is that executions effectively help validate the workflow (although there can be complexities[56]; as with computer programs, executing the full set of possible reactions can be non-trivial). Without the ability to execute, there is no easy way to verify their correctness.

2.2.2.4 Nestings

Workflows can become very large and unfathomable. A solution to this is to use nestings, and isolate sections of a workflow into separate, lower-scope workflows. This will be of particular relevance to our research, as certain proposals and deliberations may relate to a particular level of nesting. Additionally, nestings can potentially be exploited to contain specific deliberations, isolating them from agreed parts of a workflow.

2.2.3 Current Workflow Research

Much current research focuses on the adaptation of executable workflows [5]. This consists of making workflows dynamically adjust to respond to the inevitable changes that would otherwise "derail process execution"[57]. More specifically, the need for workflow adaptation arises from two factors [58]: a changing environment (e.g. expanding business activities) or technical advances (e.g. replacing a software component). Workflow adaptation is often called exception handling[58] in the literature, highlighting the similarity between workflows and programming languages with regard to resilience to unpredictable change. Kammer[57] suggests that rich integration with computer-mediated-communication tools may facilitate workflow adaptation. Therefore our vision of facilitating the participation of all involved parties in the refinement of the workflow is expected to be beneficial with regard to the issue of adaptation.

In order for adaptation to occur, a workflow language needs to be extended to permit the description of its variations. Recent research in this area has led to the specification of the VxBPEL[59] language, an extension of BPEL. Its typical application is self-adaption of a Web service orchestration, for example to allow automatic reconfiguration in response to a changing environment such as service failures or Quality of Service requirements—akin to an autonomic system. Our goal is significantly different and is not focused on execution, but is instead concerned with encoding variations for the purpose of supporting deliberations. However some of VxBPEL's research findings can be useful in specifying a means to encode variation in a BPMN workflow.

Other current research areas, closely related to the aforementioned topic of adaptation, concern aspect-orientation[60]. Aspect orientation is an approach that aims to contain aspects that tend to be scattered throughout the design, on the basis that a single locus facilitates adaptation. The AO4BPEL[61] and AspectBPEL[62] languages are two further extensions of BPEL that have been introduced to support an aspect-oriented approach. Such approaches are significant to our research objective, as deliberations can potentially relate to topics which may not be constrained to isolated sections of a model but instead affect large swathes of the workflow.

Workflow patterns[44] have been defined in order to compare the expressiveness of workflow managements systems. They are broken down into categories called perspectives, which include control flow, the data perspective, and the operational perspective. Those patterns and evaluations of current process languages are being maintained on a dedicated website[63], a rigorous basis for comparison which makes it easier to select the most suitable workflow language for a given problem area.
Workflow patterns, having allowed a thorough analysis and understanding of the various limitations of existing workflow languages, have led to the conclusion that Petri nets[64], which are the basis of a number of workflow languages, need only be extended by a small number of constructs to support all known workflow patterns. As a result, a new workflow language named YAWL[65] was introduced. This language was specifically designed to draw on the lessons learnt from workflow patterns, and is based on Petri nets with three extra constructs. YAWL is therefore concise yet highly expressive – it facilitates all known workflow patterns in a simple and intuitive manner –

and its open-source engine and toolset make it a very appealing choice. Its downside in comparison with BPEL, however, is that YAWL has so far been deployed in much fewer real-life applications and does not enjoy the benefits of being maintained by an established standards body.

The developments mentioned above (exception handling, patterns, aspect-oriented design) suggest a clear parallel between workflow languages and programming languages. It is interesting to note that efforts are being made to express workflows using scripting languages. The Apache Software Foundation, working on the BPEL4Coders initiative, has proposed a JavaScript-style workflow language called simBPEL[66]. Meanwhile, academics have designed BPELScript[67]. IBM also contributed to the BPELJ[68] standard to include Java code snippets in BPEL workflows – an initiative which wasn't welcomed by everyone as it arguably breaks the intent of a BPM approach[69]. Yet these multiple forays into programming support have taken place because of a demand created by practical issues. Implementing even a basic BPEL-WS orchestration of web services can be done much more concisely in code than in an XML language such as BPEL. Graphical workflow editors require plug-ins for development environments, and incompatibilities, incomplete support of certain features, and the general unwieldiness of these plug-ins can make their use unproductive in comparison with textual code. Of course, a common distinction between workflows and programming languages is that workflows are concerned with orchestration, or programming in the large[70], whereas programming languages are generally concerned with a much finer level of detail, or programming in the small. However many programming languages and scripting languages, while suited to the small, are also capable of dealing with the large, and indeed the "programming by intention" practice encouraged by some Agile developers[71] consists of writing orchestration-style methods and then gradually more specific and lower-level methods.

2.2.4 Software Ecosystem

The business process workflow software ecosystem is well developed, with a mixture of proprietary and open-source offerings, and workflow standards are widely supported.

Consequently we can extend existing technology rather than build from the ground up. By using standards, our software implementation allows us to rely on the wider ecosystem for experimental purposes – for example, we can design an executable workflow in an experiment and then make use of an existing solution to test its execution.

2.2.4.1 Business Process Management Suites

Business process management suites are end-to-end software solutions that allow organizations to design and deploy workflows. As the BPM acronym is shared by Business Process Management[49] and Business Process Modelling, many software vendors favour the use of the term "integration" rather than "management" in their marketing. Commercial suites include Microsoft Biztalk Server [72], IBM WebSphere Business Integration Server Foundation[73], Oracle WebLogic Integration[74] and BizAgi BPM Suite[75].

2.2.4.2 Service Oriented Architecture Components

The rise of Service Oriented Architecture in the software industry has led to the development of a vast array of commercial and open source workflow engines. These include Apache ODE, jBPM, the Open Business Engine, and Microsoft Internet Information Server (supplemented with a Microsoft workflow toolset named 'Dublin'[76]).

Microsoft's .NET Framework includes the Windows Workflow Foundation platform[76], which supports the development of distributed software applications using workflows. It is a software development framework designed to resolve distributed computing issues, a specialization which makes much of it irrelevant to the business modelling scenario which we are concerned about.

2.2.5 Design for Collaborative Workflow Editing

2.2.5.1 Collaboration Frameworks

The emergence of Web 2.0[1] has allowed web collaboration infrastructure to develop and mature, with notable recent innovations such as Google Wave[77][78].

1. Google Wave, in spite of it being discontinued as a dedicated Web application following an unsuccessful launch[79], has been used to support two competing workflow modelling implementations; one by SAP[80] and the other by Itensil[81]. Their focus is on concurrent, real-time editing, showcasing the synchronous collaboration capabilities of Google Wave.
   In our context, as the contributors to a workflow are also typically its users, individual refinements are likely to continue over time. This is akin to Wikipedia[82] article editing, in which a visitor—a subject matter expert or merely someone who happens to possess a relevant piece of knowledge—who reads an article would amend it immediately should she notice an inaccuracy or an improvement that can be made. As a result of this, a synchronous collaborative approach is not always appropriate given the greater importance of delivering good usability for asynchronous editing.

2. BPMNCommunity.org [28] is a platform specifically developed to host workflows modeled using the Oryx editor, described below. BPMNCommunity.org is akin to a wiki. All users are allowed to edit all the workflows available on the site; there is no access control. It features global-scope voting and discussion support, but it is not possible to vote or comment on a specific part of a workflow or on a specific change made to a workflow. This restricts the scope of deliberations, and our research demands support for debate at a finer level of detail. Basic workflow change visualization functionality, a key requirement for distributed collaboration, is implemented.

2.2.5.2 Graphical Workflow Design Libraries

Numerous reusable workflow editing libraries are in existence. They include Integrated Development Environment plug-ins, such as Eclipse plug-ins AgilPro, Embarcadero and Soyatec eBPMN. However a number of libraries specifically designed for Web applications are also available; they include the following:

1. Oryx[23][83], an open-source academic web-based editor. Its supported standards include BPMN and XPDL.

2. HOBBES[84], an academic web-based BPEL editor designed for scientific work-

flow management.

3. Lombardi Blueprint[85], a commercial web-based editor.

4. mxGraph[86], a proprietary client-side diagram editor which can support workflow diagrams. It is however not specifically designed for workflows and, unlike Oryx, there currently is no built-in support for workflow standards such as BPMN and XPDL.

These solutions are built on top of different front-end technologies. In their research for the design of the HOBBES collaborative BPEL editor[84], Held and Blochinger identified potential development platforms for the design of a workflow editor as a Rich Internet Application: Client-side scripting/AJAX, JavaFX, Silverlight, Adobe Flex. They chose the latter after discarding the client-side scripting approach as it suffers from complications caused by browser incompatibilities.

The authors of Oryx however, made the choice of client-side scripting, with the use of SVG[87] for graphical representation. SVG, as a vector graphics technology, is particularly well suited for workflow rendering, although it also has its own browser incompatibility issues [88], and is not at present well supported by the leading Web browser, Internet Explorer, as highlighted by Tim Berners-Lee[89]. This is however an issue for the real world – for our experimentation we can prescribe a specific browser (Oryx's recommendation is Firefox 3) and need not concern ourselves with incompatibilities.

2.3 Summary

We began this review by examining the topics of online communities, self-governance and online collective action. We looked at the essential factors that influence the success of a Web application in establishing and sustaining a thriving online community and we examined how corresponding principles have been applied to facilitate online collective action, with a particular focus on Wikipedia.

We found that research into applied business process modelling calls for more inclusion during modelling, partly to reduce the perception that business processes are imposed to rank and file workers by leadership, and partly to unlock the knowledge of the people involved in carrying out the tasks and who have specialist knowledge of the work. In essence, we made the case for a wiki approach to business process modelling.

We then delved into the technical aspects of workflow systems, which are used in a continuum of settings ranging from the loose definition of business processes to executable workflows driving business activities and human tasks to the orchestration of Web Services and grid computing. A wide range of workflow languages is in existence, and workflow pattern research has emerged to provide a basis for their comparison. YAWL is a concise and expressive workflow language specifically designed as a result of this research. We then reviewed existing online workflow editors and determined what technologies have been exploited to address their requirements.

Chapter 3

Design

Our review of the State of the Art puts us in position to make informed choices towards the design of a solution supporting collaborative workflow editing for online collective action. In this chapter we isolate the principal requirements of a proposed system and discuss the reasoning behind major design choices. We then develop a more detailed understanding of this system by discussing the key concepts of its design.

3.1 Functional Requirements

Our review of the State of the Art has highlighted the crucial importance of self-governance, sociability, rich communication, visibility and usability for members of online communities. These aspects shape our functional requirements:

3.1.1 Self-Governance

As discussed in section 2.1.1, the approach to online community self-governance currently observed in most parts of Wikipedia is to provide unrestricted access to all users while entrusting the community with the responsibility of socially enforcing its own rules. This will also be our approach—the online community should have the freedom to evolve its own rules, and the application should not be unduly prescriptive.

Consensus building will be supported by providing unrestrained deliberation support accompanied with voting.

3.1.2 Sociability

The application should provide adequate sociability support. Rich communication should be central to the application, and users and their contributions should be highly visible. For this we rely on feeds with a miniature picture of the user.

Users should form a reputation over time. We provide a user page with a feed of contributions and statistical data showing the number of contributions made.

3.1.3 Usability

The application should be intuitive and enjoyable to use. We use drag-and-drop editing and simple, highly visual features such as change visualization and replay. Interactions should be instantaneous—we use AJAX for a rich Internet application experience.

3.1.4 Visibility

A history of amendments should be available, detailed (who, what, when), and change visualization features should help isolate which workflow elements have changed.

Contributions should not grow over time in a manner that becomes unmanageable or makes information difficult to find. Therefore the data should be organized in a manner which leaves all historical data accessible without polluting recent data.

3.2 Design Choices

3.2.1 Project Scoping

Our review of the State of the Art allows us to narrow the scope of workflow modelling to suit our application domain.

In particular, we have seen in section 2.2.2.3 that a distinction can be made between informal and executable workflows, the latter of which include data flow information and require a greater level of detail and precision. It would be significantly more difficult to evaluate the design of executable workflows: Doing so would require participants with a high level of skill in workflow modelling and who, given the greater complexity of the models, who would be willing to volunteer more time for the evaluation. For this reason, workflow execution was considered out of scope, and features such as exporting workflows and enforcing valid syntax are not required.

3.2.2 Workflow Language

As we have seen in the State of the Art, a wide array of workflow languages exists. For this project, given the low importance of execution support, two of the languages we identified were considered:

1. BPMN[90]:
   Well-established industry standards supported by a large software ecosystem.

2. YAWL[65]:
   A more academic language, stemming from patterns research, which is designed to be concise yet highly expressive.

Should this be a commercial project, BPMN would have been an attractive choice. However for the purpose of this research, the concise grammar of YAWL was found to be a strong advantage, while its lagging position in industry was largely inconsequential. Our choice was therefore to adopt the YAWL language. Given its relative simplicity, we were able to develop an editor offering the full YAWL grammar.

3.3 Key Concepts

3.3.1 Revision Tree

To allow a workflow model to evolve through successive revisions whilst making it possible for users to propose alternatives to a model, a versioning system had to be established.

This project proposes doing so by means of a tree structure which we refer to as the "revision tree". Each node in the tree consists of a single revision and may have any number of child nodes, which are modifications of itself. Each time a user edits and saves a model, a new node is created in the tree. Hence the model represented by a node is immutable—its workflow cannot be modified, and editing it merely creates a new child.

3.3.1.1 Referencing System

Each node in the revision tree is referenced by a series of numbers separated by colons. To generate a tree reference, one traverses the tree from the root to the target node and inscribes the index of each child node, separated by a colon. For example:

- "1" is the root
- "1:2" is the second child of the root
- "1:2:1" is the first child of the second child of the root

These references can get unwieldy in the case of a long series of consecutive edits. Hence a shorthand notation is proposed, in which any repeated series of index i is replaced with the pattern $i \times n$, where n is the number of repetitions, whenever n is greater than 2. For example, the shorthand for "1:1:1:1:3:2:2:2" is "1x4:3:2x3". In practice, we expect to see the index 1 repeated most frequently, as branching tends to be less frequent than consecutive edits.

3.3.1.2 Current Version

Systems that do not allow branching or proposing pending changes, which we hencetoforth call *single-tracked* systems, implicitly determine what the current version is—it is the current one. Our revision tree and its support for branching, which in contrast we call *multi-tracked*, lacks such an implicit definition of a current version. Hence the current version can only be explicitly declared through user interface controls. We provide a "Make current" button on the model view pages and display the current node using special signage on the version tree. We will discuss this signage, shown as figure 5.4(b), in chapter 5.

While not strictly necessary, the concept of a current version is useful to determine which of the version currently represents the workflow. On the application home page, for instance, choosing a workflow opens its current version. When branching occurs, i.e. more than one version of the workflow can be chosen, the online community may deliberate on which version should be attributed the status of current version. The voting and commenting features provided by our prototype are intended to facilitate such deliberations.

3.3.1.3 Historical Versus Leaf Revisions

Some scenarios require processing certain nodes in a specific way, and we need introduce specific terminology. Nodes in the revision tree that have at least one child node, i.e. parent nodes, are called "historical revisions". All other nodes, i.e. leaf nodes, are called "leaf revisions".

3.3.2 Data Permanence

Any contribution made is stored permanently and cannot be deleted.

This raises the question of the impact that progressive growth may have on usability. In Wikipedia "talk pages", historical deliberations can be gradually removed to keep the page's size manageable. In our application, however, updates are isolated from the rest of the data due to the use of the revision tree. Consequently, the growth of data merely results in the addition of new nodes in the revision tree, thus allowing us to keep historic data available in separate nodes without compromising the manageability of leaf revisions.

Not permitting deletions is a feature. It ensures the full traceability of contributions: the history of the model and all contributions can be retrieved. Additionally, this aspect guarantees the stability of the revision tree and its referencing model. Tree node references (e.g. 1:2:3) are immutable and unique because any existing node cannot be deleted.

3.3.3 Nestings

Sub-processes are separate workflow documents, which have their own revision tree. The only difference between sub-processes and processes is that the former are not displayed on the home page.

The definition of a nesting is akin to a hyperlink, pointing to a specific version of the subprocess.

Should the nesting link require an update (after a new version of the target sub-process has been added for example), a corresponding new revision of the source model is required, since revision nodes are immutable by design (to maintain a full trace of changes). However, automatic updating of leaf revisions can be considered; for example, should a contributor amend the current version of a sub-process, a pop-up window could propose updating of all models that were linking to it.

3.3.4 Change Tracking

The workflow editor maintains a change list, adding details of any modification made. Edit-time change logging can be used to implement undo and redo, but its principal objective lies in the visibility features. The change list is saved with the revision, and is used in the view pages in the following ways:

1. Textually
   A description of the changes is displayed, both in summary and in detail.

2. Graphically
   The changes are visualized using a descriptive scheme (colour-coding and display effects giving a distinct representation of each type of change.)

3. In motion
   The changes are replayed as they actually happened.

3.3.4.1 Classification of Changes

As we have seen in section 2.1.2.4, Wikipedia allows users to distinguish minor and major updates. The change list can be used to provide automatic change classification with a much greater level of detail.

This classification is broken down into categories and types. The *structural* category is akin to Wikipedia's "major" category, and includes anything that changes the nature of the workflow. Were the workflow executable, the structural category represents changes which would have a tangible effect on execution, while in contrast, categories, *textual* and *cosmetic* are minor changes which would not. Category *external* is for changed nesting links.

The application provides a colour-coded scheme based on these categories (e.g. red for structural) to make them easier to indentify. Table 3.1 shows the complete list of changes and their categories.

Category	**Type**
Structural	Added
Structural	Removed
Structural	Connected
Structural	Disconnected
Textual	Renamed
Textual	Annotation Added
Textual	Annotation Removed
Cosmetic	Moved
Cosmetic	Resized
Cosmetic	Repositioned
External	Link Changed

Table 3.1: Categories and Types of Changes to a Model

The Connected, Disconnected and Repositioned changes are specific to the bindings that the editor creates when objects such as decorators and arcs are attached to other shapes. The distinction between Repositioned and Moved is that the former describes adjusting the position of an object inside the shape to which it is connected, whereas the latter represents moving an object on the canvas. For example an arrowhead connected to a task rectangle would be repositioned when, for instance, its attachment is changed from the left edge to the top-left corner of its rectangle.

3.3.4.2 Change Combining

The full change list can include unnecessary repetitions (e.g. successive moves) or changes that cancel each other out (e.g. adding an element and then removing it). This is increasingly more likely to occur when the changes for multiple revisions are added, as we will discuss in section 3.3.4.3.

This introduces the need to combine changes in order to synthesize a simplified list of changes, which can then be used irrespective of the change representation used—textual, graphical or in motion. A discussion of the algorithm is beyond the scope of this dissertation.

3.3.4.3 Comparing Based on Changes

The change lists provide an effective, simple way of establishing the differences between any two versions of a workflow.This is applied in the Compare page that we discuss in section 4.3.4.

The principle is to walk the revision tree from the source revision to the target revision and collect the change lists found on the way. Should these revisions be on different branches, this would require traversing the tree in two directions: upward until a common ancestor is reached, then downward until the target is reached. The change lists collected while traveling upward need to be inverted. Their order is reversed, and an opposite change is synthesized and put in their place—for example an "Add" becomes a "Delete", and a "Move" from *p1* to *p2* becomes a "Move" from *p2* to *p1*.

Having completed this tree walk, the resulting list of combined changes (also simplified as per section 3.3.4.2 above) describes the differences between the source and target revision. This is a method for "diffing" which does not actually require comparing the two documents.

3.3.4.4 Other Uses of Change Tracking

The change list, assuming that each entry successfully describes all aspects of the change made to the model, can easily be exploited to reconstruct the model in any of its representations (SVG format for rendering and YAWL format for exporting). The Compare feature makes use of this fact during replay to transform the model from its starting state to its final state, using the change lists of each node in-between them (possibly on separate branches) to progressively transform and reconstruct the target model.

Should a YAWL file export feature be implemented, it would be trivial to do so by exploiting the change logging functionality. Whenever a change is recorded in the editor, a corresponding change could be made to update the YAWL model (as single method call to a YAWL update method, passing the change object as argument), thus guaranteeing a synchronized YAWL representation which can be saved and exported at any time. This is much simpler than attempting to translate from one format to another.

Storing both the change list and the model in the database is actually a form of data duplication since the model can easily be recreated from the change list, as exemplified by the Compare page's replay functionality. However, reconstructing models from change lists would be less efficient than storing them, which would be compounded by the frequent use of model data arising from the display of multiple models on a typical page (full model view, previews of the nestings, and thumbnails in the feeds). Given the low cost of data storage and the expectation of fast page loads, this kind of data duplication makes practical sense.

Change tracking was one of the interesting discoveries of the design and implementation phases. It began as an isolated feature intended to improve on the Wikipedia "minor" indicator but grew to unlock other features, from visualization to replay and, as mentioned above, showed its potential place at the core of the data model. The importance of change tracking was an unanticipated aspect that only became clear as development progressed.

3.3.5 Feeds

Feeds are ordered collections of all events relating to a subject, which may be a revision, a user, or a workflow.

Feeds provide visibility on all interactions with the system while providing users with an opportunity to comment and debate any such interaction, as we will discuss in section . Feed items can represent a wide range of events, which are listed in table 3.2.

3.3.6 Notifications

Unlike feeds, which provide a historical view of activity (including deliberations), notifications are transient messages whose purpose is to draw attention to the particular status of the workflow and guide the user. The concept of notifications is borrowed from the Wikipedia templates discussed in section 2.1.2.4. For example, versions marked as

Feed Entry Types
Comment Added
New Model Added
New Sub-Process Added
New Version Added
New User Account Added
Update of Linked Nesting Added
Child Revision Added
Sibling Revision Added
Rating Added/Updated
Version Marked/Unmarked as Draft
Version Marked/Unmarked as Dead
Version Made Current
Version No Longer Current
Nesting Link to This Version Added

Table 3.2: Types of Feed Entries

dead branches are displayed with a notification explaining the significance of this marking. Notifications are displayed at the top of the view pages; this is illustrated in section 4.3.3.

3.3.7 User Pages

Each user account comes with a matching user page. Wherever a user name is displayed, it is hyperlinked to their user page.

Users can personalize their page with a picture and a biography, which gives some extra scope for sociability. Although frivolous, personalization of user pages and signatures is popular among Wikipedia's most prolific contributors, and allowing for some expression of individuality may strengthen sociability.

The principal function of the user pages, however, is related to visibility and reputation, through the complete user feed and the statistics which cover all measurable aspects of their activity (e.g. the total number of revisions they contributed, how many have since been marked dead branches, etc.)

3.3.8 Support for Deliberations

3.3.8.1 Commenting

To support discussions, a wiki and a forum were considered. Both of these choices, however, would not have integrated seamlessly with the feed. Consequently, the approach of the commentable feed was adopted. This is akin to Facebook's "friend feed"[91] but with support for recursive commenting, i.e. the ability to comment in response to another comment, as in an Internet forum.

Commenting is thus tightly integrated with the feed (users can comment on specific events in the feed, such as "This revision was made current by Bob"), and is achieved by appending a link labeled "Add a comment" at the bottom of each feed item.

3.3.8.2 Voting

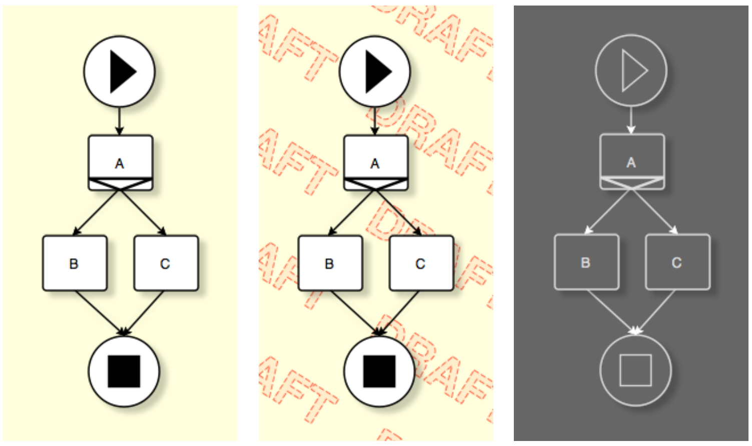
Similarly, different options were considered for voting. A system that supports the election of the current version, based on the outcome of public voting, was considered. However such a feature introduces a number of issues, such as how to determine when voting should end (after a set period of time? until a threshold of votes has been expressed?) and how to deal with multiple demands for election (prevent calling another vote if one vote is ongoing?) Another concern is that such a system may prove to be too rigid for the more informal usage scenarios. Consequently, the approach of openness coupled with the social enforcement of rules was adopted. Voting functionality is provided for every revision, in the form of a scale ranging from 1 to 5. Voting is public, with individual votes displayed in the feeds and the voting summary (number of votes and average) displayed below the voting control. Users can take any action regardless of the voting, and it is up to the community to establish its own rules regarding how voting should be used.

3.3.9 Support for Visibility

3.3.9.1 Attributes

While all the models in the revision tree are immutable, the use of metadata, which can be changed at any time, allows users to mark revisions to draw attention to conclusions that have been drawn. Users can mark a revision as draft to indicate that the revision is work in progress, with the same objective as that of Oryx's sketchy style discussed in section 2.1.2.4: making this fact more visible encourages the appropriate level and type of interaction from participants.

Similarly, users can mark a revision as a dead branch to indicate that the revision should be considered discarded and that no further work should be done from it. The draft and dead branch status can be conveyed through the use of visual styles that are immediately recognizable, as illustrated in figure 3.1.



(a) Unmarked (b) Draft (c) Dead branch

Figure 3.1: Proposed visual styles to convey model status

3.3.9.2 Feeds and User statistics

Feeds shown across the data set provide for full visibility and tracing of the contributions and of interactions of all kinds (e.g. including the marking of attributes as mentioned above), sorted by date and time. User statistics facilitate the qualitative and quantitative interpretation of user contributions, supporting the emergence of user reputations.

3.3.9.3 Change Lists and Visualizations

The full tracking of changes and the three modes of representation (textual, graphical and in motion), all of which follow the same colour-coding scheme, complete our visibility features.

3.3.9.4 Presence

A feature designed to show the real-time presence of a user was considered. This can be achieved by means of “online now” cues near the names of users, or through a dedicated control showing the list of active users. Web applications being essentially connection-free, the criteria for deeming a user’s online status can be based on a measure of time since the last round-trip, possibly enhanced with timer-based heartbeat notifications embedded in client-side script which trigger regular queries to the server. This feature was given a low priority, in part given the difficulty of incorporating it into an evaluation, and was not implemented.

Chapter 4

Implementation

In this chapter, we examine the more practical issues affecting the implementation of the prototype. We justify our implementation choices and present the resulting system that we developed.

4.1 Chosen Technologies

4.1.1 Client

Implementing web-based drag and drop editing of a graphical document such as a workflow requires adequate Rich Internet Application (RIA) client technology. We discussed in section 2.2.5.2 some of the different technologies that have been successfully applied by the authors of comparable solution. In light of the successful application of SVG in the Oryx editor, our choice is SVG[87], the W3C[92] standard for Web-based scalable vector graphics.

4.1.2 Server

As a Rich Internet Application, much of the functionality resides on the client side, and the role of the server is little more than performing data storage and retrieval, be it in response to AJAX queries or in serving Web pages with dynamic content. Much of the dynamic content of Web pages in fact consists of raw data inserted into JavaScript code blocks (typically encoded in JSON[93]), which is then interpreted by the client-side script to generate Web page content from it.

In consequence, any server-side technology would be adequate and what motivated this choice was purely practical. We decided to use PHP as it is the most readily available server technology for students, being available on the web server that hosts student pages. Other choices would introduce more distracting server procurement, configuration and deployment issues, which could potentially complicate the conduct of the evaluations.
Java came a close second, given the Java implementation of the Batik SVG Toolkit[94], which is not available on any other platform. Batik would permit the implementation of PDF or image file downloads from a workflow, for example.

4.1.2.1 Data Store

For practical reasons, Sqlite[95] was the chosen database system. Being file-based and serverless, it requires no database server configuration and makes resetting the sample data at the start of each evaluation as easy as overwriting a file. It is ideally suited to the low-scale and low-concurrency conditions under which the prototype was developed and used. However care was taken to isolate all data access from the server-side code to make it easy to replace Sqlite with another database system should the need arise.

4.1.3 Frameworks

The use of Web frameworks and Content Management Systems (CMS) was considered. Such frameworks, being robust and maintainable, are most helpful when supporting "real-world" Web applications. The case for their use in a research prototype is weaker, however. The learning curve and the risk of unanticipated integration issues contributed to the decision not to use such frameworks.

On the client side, however, the prototype[96] JavaScript framework and its extension script.aculo.us[97] were adopted. These frameworks facilitate client-side web development, particularly AJAX features.

Recent JavaScript frameworks dedicated to SVG document manipulation and cross-browser SVG support were also considered (jQuery SVG[98] and Raphaël[99]), but were not adopted due to uncertainties over the level of control they allow.

4.2 Editor and Viewer Implementation

SVG is an XML-based document format that can be manipulated through its Document Object Model (DOM). We define standard shapes of each of the YAWL syntactic elements (rectangles for tasks, circles for conditions, arrows for arcs, are more complex shapes for decorator constructs such as joins, splits, multiple instance indicator) which we include in the toolbox frame in the editor. When shapes are added through drag and drop, the corresponding shape from the toolbox is duplicated, a Globally Unique Identifier (GUID) is assigned to it, and the new shape is appended to the SVG canvas element. Any interaction with a shape, such as moving, resizing, editing its text, is done by manipulating the corresponding shape through the SVG DOM. Finally, saving is done by persisting the XML document to the database. Hence editing is essentially graphical in nature, except for a limited amount of non-graphical information that is added to the SVG document where necessary. Such added information includes the GUID and the anchoring attributes, and is appended by means of dedicated attributes in a private namespace—being an XML format, SVG is of course extensible in this manner.

The anchoring attributes (*anchor* for attributes, *anchorhead* and *anchortail* for lines) determine which shape, if any, the corresponding element is bound to. To create such a binding, the attribute is assigned the GUID value of the target element.

4.2.1 Object Model

An object model reflects the state of the document, facilitating manipulation of the document in an object-oriented manner while enforcing the required rules. For example, the CBinding class represents a binding, information which is also encoded in the SVG document as we discussed above. A collection of CBinding objects provides an effective way of managing bindings, and is much more practical than relying on the data encoded in the SVG DOM.

Likewise, the CItem class and its hierarchy of derived classes such as CTaskItem, CLineItem etc. represent the different types of shapes encoded in the SVG document. These classes facilitate the definition of specific behaviour for each shape in a typical object-oriented manner—variation of behaviour is abstracted out into subclasses. The object model includes a number of other classes that we will not detail here.

The object model is reconstructed when a model is loaded: The SVG XML content is examined and each shape triggers the reconstruction of a corresponding CItem object (a factory method pattern ensures that the appropriate subclass is used).

The graphical workflow code supports two modes: "edit mode" for use in the editor page, and "view mode" anywhere a workflow is displayed (which can be in thumbnail size or in actual size). Regardless of the mode, the object model is constructed. It is required in view mode because the change visualization features make use of it to manipulate the SVG document. For instance, highlighting a shape is done through the object model, causing the relevant SVG element definition to be appended a CSS[100] class attribute comprised of an SVG filter[87] that adds a red glow around the shape. The need to construct the object model is particularly true in the case of replay-style visualizations, which modify the workflow in a manner similar to the how the user edited the workflow. Indeed, accurate replay can only be guaranteed if the same code execution occurs: the side-effects of every user action, which may depend on changing states, are reproduced. For example, connecting an arc to a shape and then moving this shape causes the arc to change indirectly, which is enforced by the object model.

4.3 Features Overview

4.3.1 Home Page

Figure 4.1 shows the home page. A bar at the top, present in all pages, allows users to log in and out. Previews of the current version of the workflows stored in the database are displayed in the center; on this figure, there are only two existing workflows, not including sub-processes which are not displayed on the home page.

4.3.2 Editor

Figure 4.2 is a screen capture of the prototype's editor. The toolbox on the left contains all the constructs that make up the YAWL syntax. These can be dragged and dropped to add them to the model. Elements can be moved and resized (via dragging), deleted (by clicking the element to select it then pressing the delete key on the keyboard) and renamed (by double-clicking it) and, in the case of arcs and decorators, connected to

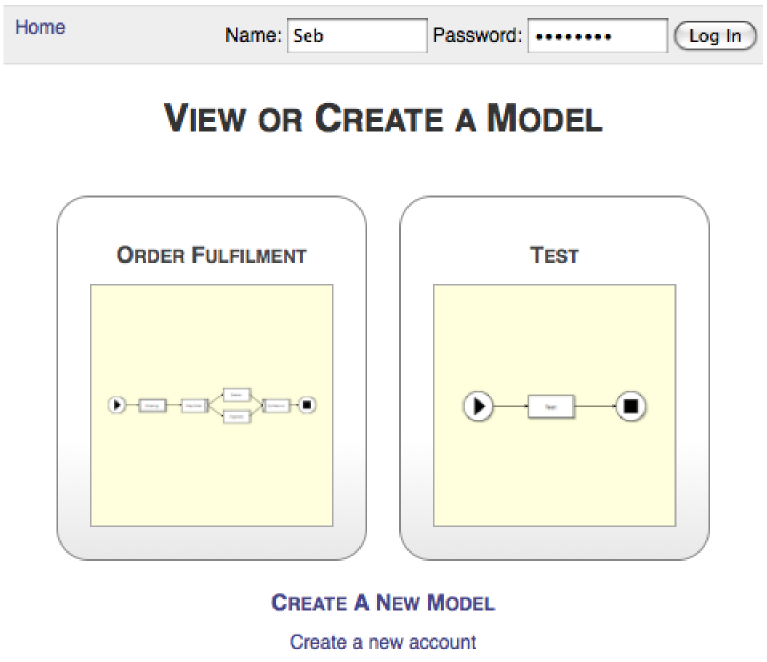


Figure 4.1: Home page

other shapes by dragging them onto them.

Figure 4.3 show a nesting chooser UI control. For each nesting in the workflow document, a corresponding nesting chooser control is created and displayed below the model in the editor.

4.3.3 View Page

Figure 4.4 show a view page, displays a workflow version, which happens to be marked as draft.

At the top, we see the title and revision number, followed by a notification frame.

On the left is the revision tree, with the node highlighted in red representing the position of the version being viewed. The current version is marked as a home icon—here it happens to be the same version as the one viewed.

At the right side of the page, a hyperlinked preview of the nested workflow is displayed. It corresponds to the nesting titled "Ordering" on the model—nestings are represented as rectangle with doubled edges.

Below the model are controls to edit, mark/unmark as draft or dead branch, and

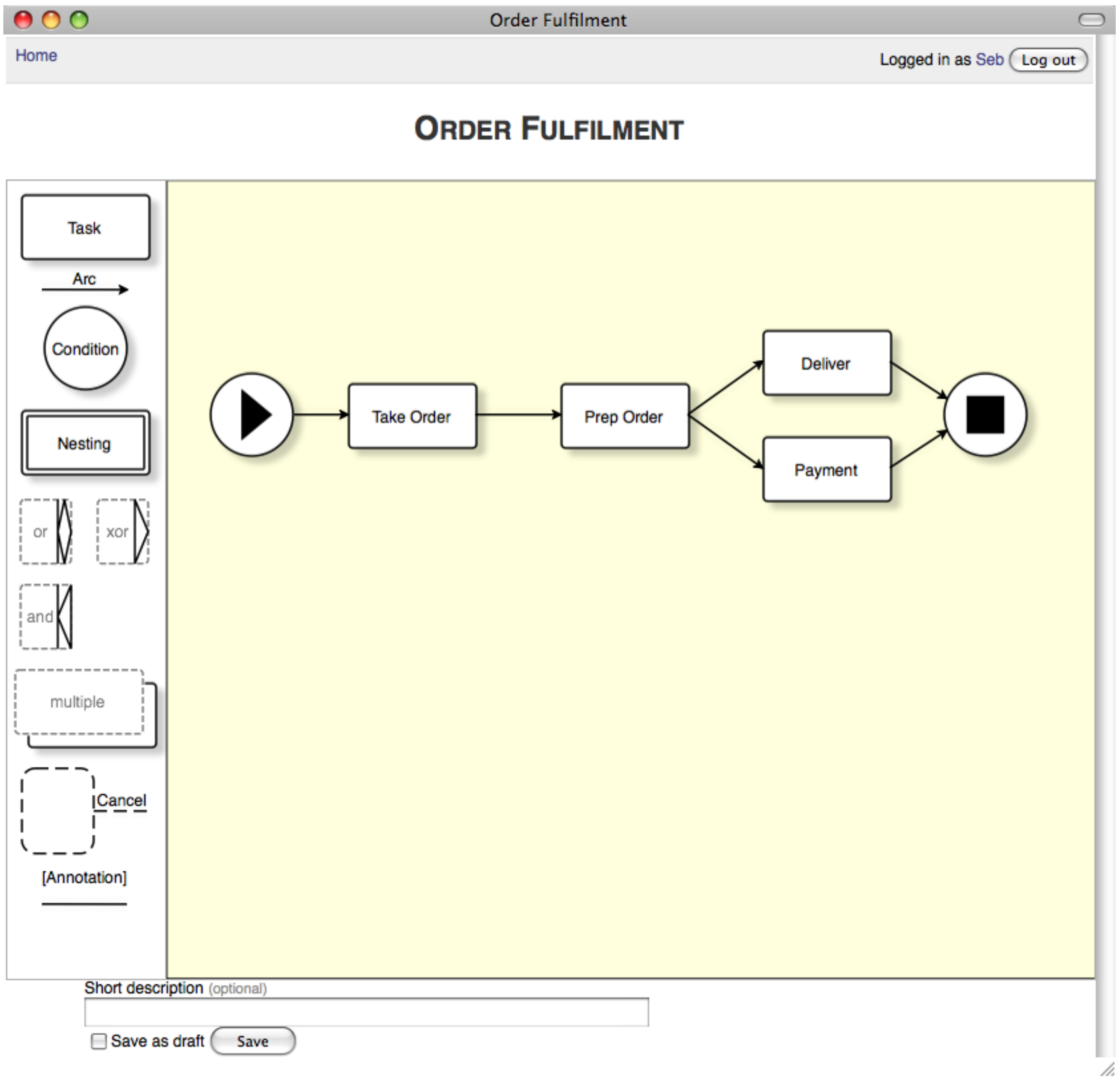
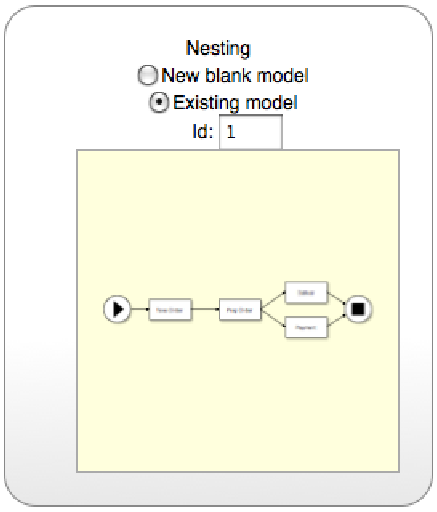


Figure 4.2: Editing a workflow


Nesting

New blank model [✓ Existing model]

Id: 1

Figure 4.3: Editor UI for a nesting

vote.

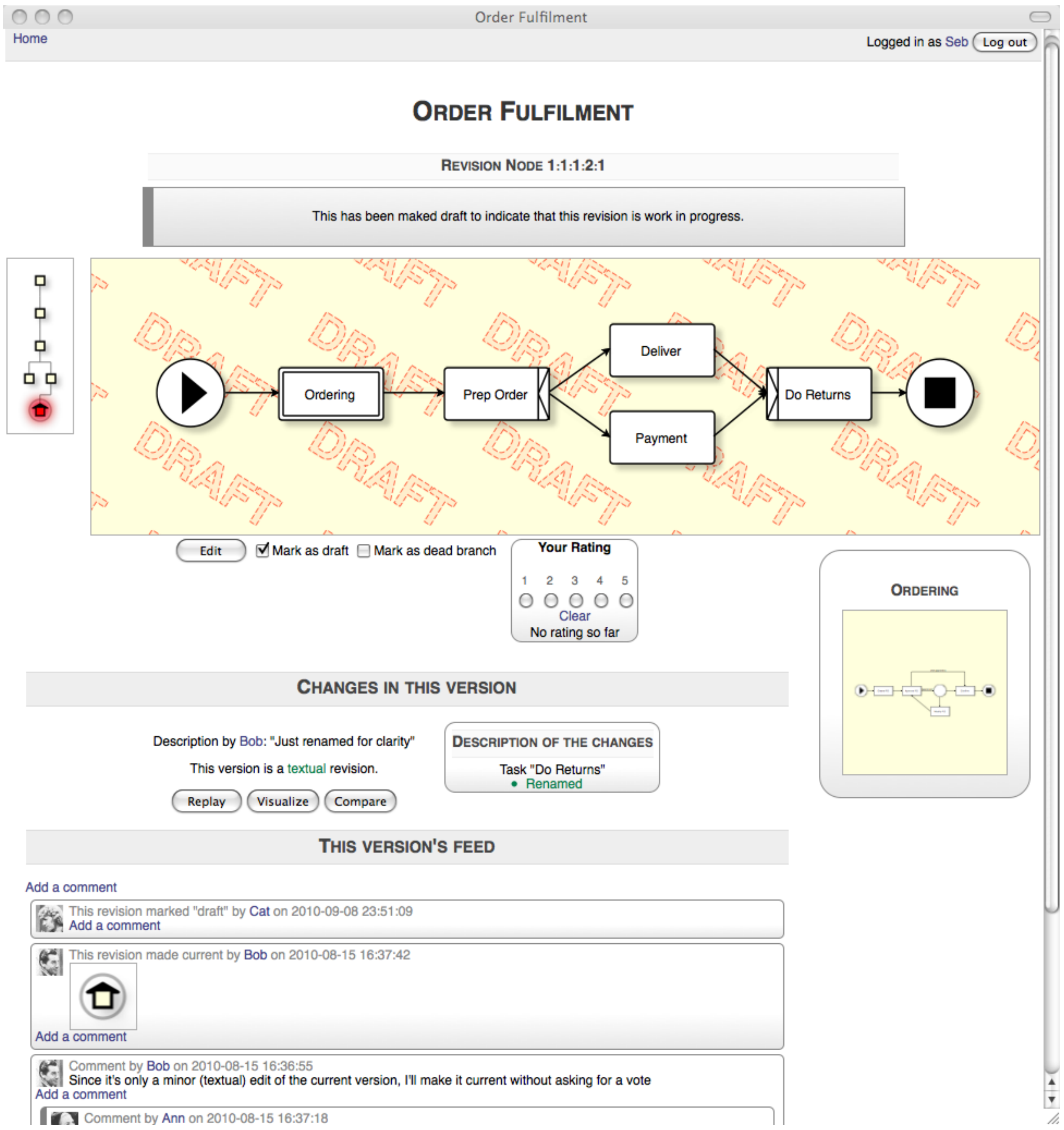
Below this is the list of changes, displayed in textual form—in this case there is just one change, a text change, displayed in green. To inform users of the importance of the change, the categories of the changes is displayed; in this case we see “This version is a textual revision”. “Replay” and “Visualize” buttons let users visualize these changes, and the “Compare” button takes the user to the compare page, which we will describe in section 4.3.4.

At the bottom is the feed. It is truncated; should we scroll to the bottom, we would see a link which, when clicked, displays a further feed listing all events relating to all other revisions of the model.

4.3.4 Compare Page

Figure 4.5 shows the compare page. The revision tree in the control is clickable and allows to select the version to which the user wishes to compare the source version, which is highlighted in red. The nodes to be traversed are displayed in green—in this case, the user wishes to compare the source version to the current version, identified by the “home” icon.

Below this is the workflow. This screenshot was taken after the user clicked the Visualize button, and colour-coded highlighting identifies the changed elements. Using



ORDER FULFILMENT

REVISION NODE 1:1:1:2:1

This has been maked draft to indicate that this revision is work in progress.

No rating so far

CHANGES IN THIS VERSION

Description by Bob: "Just renamed for clarity"

This version is a textual revision.

DESCRIPTION OF THE CHANGES

Task "Do Returns"
- Renamed

THIS VERSION'S FEED

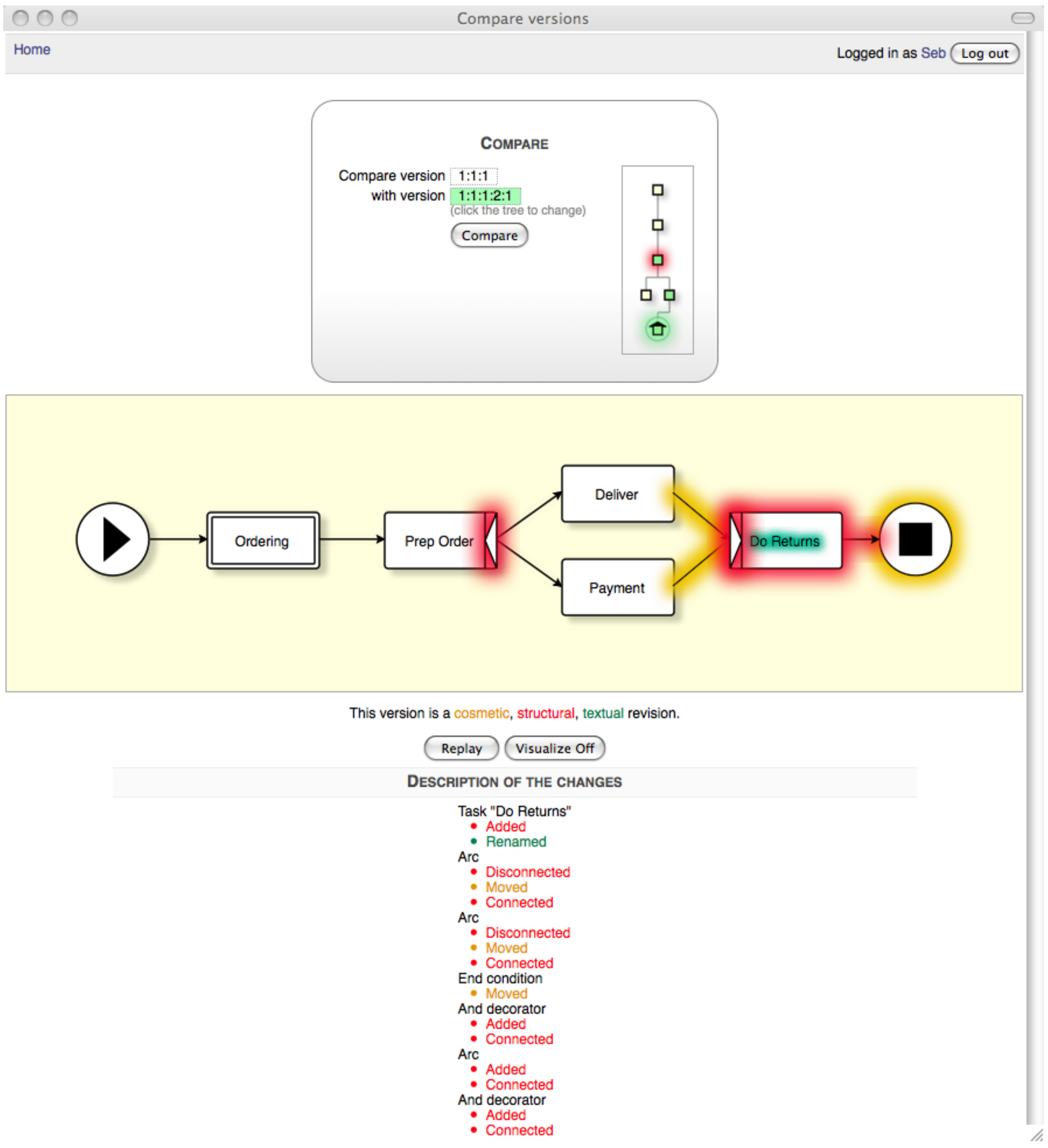
This revision marked "draft" by Cat on 2010-09-08 23:51:09

This revision made current by Bob on 2010-08-15 16:37:42

Comment by Bob on 2010-08-15 16:36:55
Since it's only a minor (textual) edit of the current version, I'll make it current without asking for a vote

Comment by Ann on 2010-08-15 16:37:18

Figure 4.4: View page



Compare version 1:1:1
with version 1:1:1:2:1
(click the tree to change)

This version is a cosmetic, structural, textual revision.

Description of the changes

Task "Do Returns"
- Added
- Renamed

Arc
- Disconnected
- Moved
- Connected

Arc
- Disconnected
- Moved
- Connected

End condition
- Moved

And decorator
- Added
- Connected

Arc
- Added
- Connected

And decorator
- Added
- Connected

Figure 4.5: Comparing two revisions

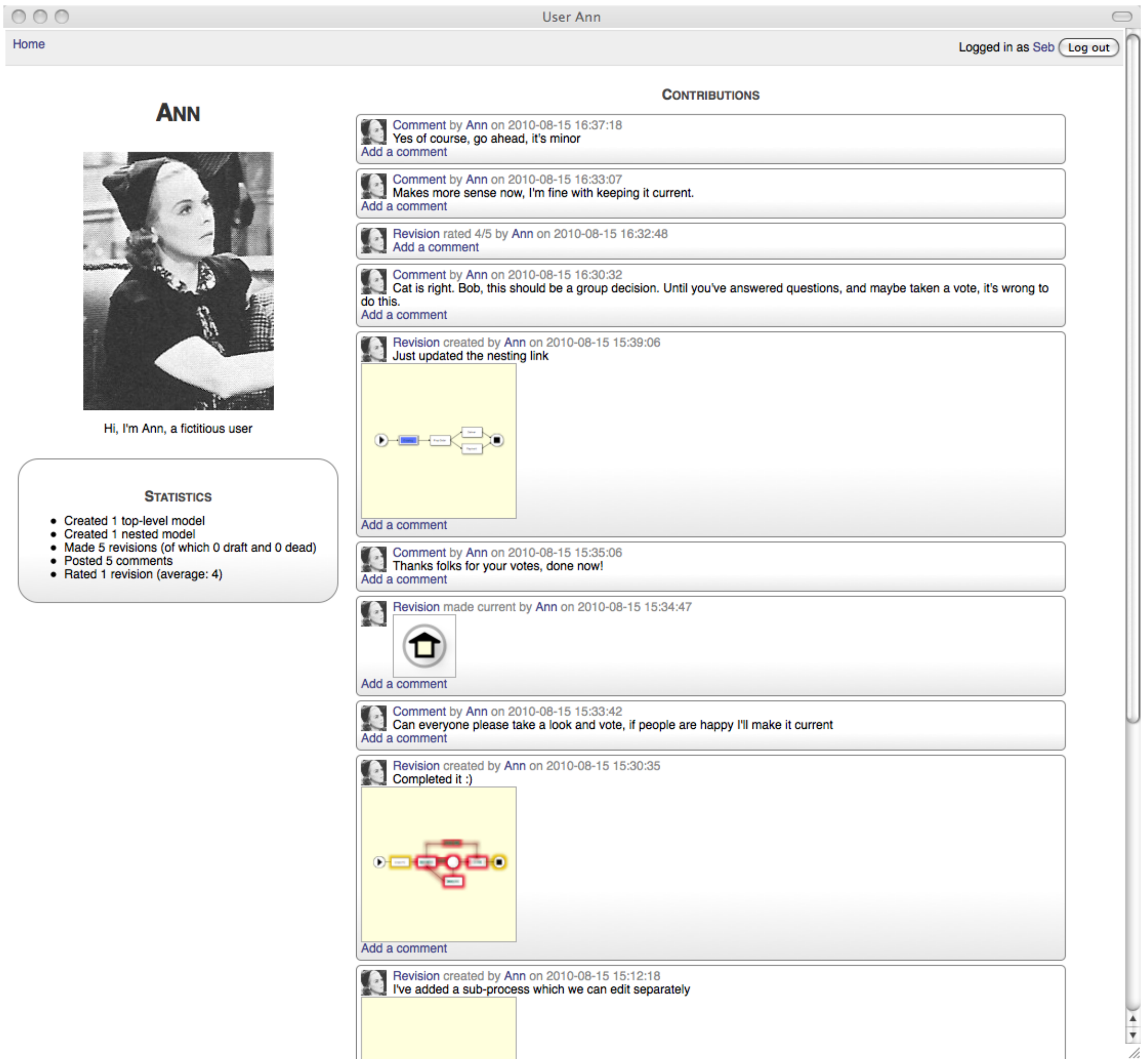
the same UI elements as the View page, a textual description of the differences is displayed alongside their categories (in this case "cosmetic, structural, textual"), with the same colour-coded scheme as used in the model's visualization, and the Replay button animates these changes.

4.3.5 User Page

Figure 4.6 shows the user page for user Ann. The profile is shown on the left and includes the user's name, a photograph (in this case, a public domain image was used for the sample data) and a short biography.

The frame below contains the user's statistics, showing eight values, all but one (the average rating) being counters.

The user's feed is also displayed and contains all the events recorded for this user sorted by date and time. Thumbnails with graphical visualizations are included in all feed entries related to the creation of a new workflow version, facilitating the visual interpretation of feed data.



User Ann

Home

Logged in as Seb Log out

ANN

Hi, I'm Ann, a fictitious user

STATISTICS

- Created 1 top-level model
- Created 1 nested model
- Made 5 revisions (of which 0 draft and 0 dead)
- Posted 5 comments
- Rated 1 revision (average: 4)

CONTRIBUTIONS

Comment by Ann on 2010-08-15 16:37:18
Yes of course, go ahead, it's minor
Add a comment

Comment by Ann on 2010-08-15 16:33:07
Makes more sense now, I'm fine with keeping it current.
Add a comment

Revision rated 4/5 by Ann on 2010-08-15 16:32:48
Add a comment

Comment by Ann on 2010-08-15 16:30:32
Cat is right. Bob, this should be a group decision. Until you've answered questions, and maybe taken a vote, it's wrong to do this.
Add a comment

Revision created by Ann on 2010-08-15 15:39:06
Just updated the nesting link
Add a comment

Comment by Ann on 2010-08-15 15:35:06
Thanks folks for your votes, done now!
Add a comment

Revision made current by Ann on 2010-08-15 15:34:47
Add a comment

Comment by Ann on 2010-08-15 15:33:42
Can everyone please take a look and vote, if people are happy I'll make it current
Add a comment

Revision created by Ann on 2010-08-15 15:30:35
Completed it :)
Add a comment

Revision created by Ann on 2010-08-15 15:12:18
I've added a sub-process which we can edit separately

Figure 4.6: User page

Chapter 5

Evaluation

5.1 Approach

A shortlist of the core research elements to evaluate was drawn up. This was done relatively early and informed the prioritization of features during the design and implementation phases, as it would not have been judicious to develop functionality that had no realistic expectation of contributing to the evaluation. The shortlist was as follows:

1. Assess the visibility of contributions (volume, importance)
2. Assess the reputation of contributors
3. Assess the sociability (allowing for communication, etiquette, rules)
4. Assess the traceability of changes and decisions
5. Assess the accommodation of deliberation and consensus-building
6. Assess the ease of use

From the shortlist, an evaluation plan was written. It became clear that some aspects of the evaluation require a sufficient body of data to be available. For example, evaluating the system's support for the emergence of contributor reputation requires enough data to be produced by distinct contributors in order to distinguish their relative levels of input. Thus the required amount of user input could not realistically be expected to be produced through the limited time spent by the volunteer participants of this evaluation. In consequence, it was decided to create a sample set of data, featuring fictitious users and containing just enough data to make aspects of the core research elements apparent.

Similarly, two-way interactions between multiple participants was not included in the evaluation plan given the limited demands that could realistically be made of participants (up to one hour, including familiarization with the prototype and filling-in the questionnaire) and the difficulty in gathering repeatable, measurable data from unpredictable interactions.

The workflow content in the sample data set is adapted from the YAWL Order Fulfillment Process example [101]. It is shown as having been authored by three fictitious users named Ann, Bob and Cat, each having a distinct persona. The use of gender-neutral identifiers such as A, B, C rather than these names (and neutral images rather than photographs of faces; we used images from the public domain for our fictitious users) was considered, as this would have had the advantage of preventing bias of any kind from interfering with the evaluation. However doing so would have interfered with the evaluation of the social features of the system, which depend on having credible human personas with distinguishable features.

The respective level of contribution of the three fictitious users is designed to be markedly different when looking at their statistics, with Ann having authored the major parts of the data and the two other authors having contributed smaller revisions. The sample data set also includes a simulated example of a user, breaking an etiquette rule. Specifically, user Bob marked a version current before consensus was reached over whether his revision was superior to a competing revision made by user Cat. These two aspects, and various comments left by the fictitious users, are expected to permit the interpretation of user reputation and give a glimpse of the potential for deliberation and etiquette.

Finally, the main workflow in the sample data set is spread over six revisions and includes one nested sub-process with three revisions, thus helping to assess the effectiveness of the system in providing for the traceability of changes and for the scoping of deliberations.

For the evaluation, ten members of the Knowledge and Data Engineering Group (KDEG)[102], a research department at Trinity College Dublin, agreed to participate. An additional participant, User 11, is a classmate who also volunteered in return for participation in his own research. All participants happened to be male, hence the use of adjective “his” or pronoun “he” in the following text does not compromise any of the participant’s identity. Subjects are referred to as participant 1 to participant 11, and the numbering of these participants does not match the ordering of appearance of the volunteers in the trial schedule, also in the interest of non-identification.

The evaluation sessions were designed to take no more than one hour each. As per the school’s research procedure, approval by the ethical committee was received before user trials commenced. Each participant signed a consent form prior to undertaking the evaluation and was informed, amongst other things, of the anonymous nature of the study.

Each session began with a six-minute introduction video covering the basic functionality of the prototype (drag and drop editor, version tree, nesting references, notifications, feeds, change log, marking as current, draft and dead branch, change visualization and replay, user page and statistics, comparing versions). The transcript of this video is included in Appendix B. This video is specifically designed to introduce the major functionality of the prototype without providing answers to any of the questions that we will be asking to participants. For example, while the video shows how to access the user profile page and shows that user statistics are found there, we do not mention that it can be used to assess the reputation of a contributor, as participants are asked in a subsequent exercise to compare reputations.

After watching the video, participants followed the instructions given in the evaluation worksheet. The content of this worksheet is including in Appendix C. It begins with a warm-up exercise during which participants create a simple workflow and revisions to add two distinct revision branches, allowing the participant to form an opinion on the usability of the editor and to experience refinement and versioning. The remaining part of the worksheet consists of a series of questions about the sample data set, which require the participants to find their way through the application and interpret the data in order to find answers to them. This first question, *“Among Ann, Bob and Cat, who most deserves the reputation of top contributor? How did you establish this?”*, requires the participant to find and distinguish quantitatively and qualitatively the contributions of the three fictitious users, possibly by comparing the statistics on their user profile pages. The next question, *“Identify one major revision and one minor revision. How could you tell that the major revision you identified is more important than the minor one?”*, encourages participants to look in more detail at some of the revisions made, either through the model feed or by browsing the version tree, and intepret the relative importance of revisions. The last question, *“In online communities, a code of practice often evolves to form guidelines on what should and should not be done. Find one instance of bad behaviour by Bob. How did you find it? How did the online community address it?”*, requires that the participant interpret the commenting and deliberations that occurred in one particular version, which they are expected to find relatively easy through any of the feeds.

The questionnaire contains 17 questions, broken into three sections. It is shown on table 5.1. The first section of the questionnaire is the System Usability Scale[103] (SUS) set, comprised of ten standard questions. This gives us a benchmark for evaluating the prototype’s ease of use, which is important for a Web application’s uptake in the context of online collective action[18]. Each SUS question is rated on a five-point scale, going from *“Strongly disagree”* (1) to *“Strongly agree”* (5). From these ratings, a SUS score ranging from zero to a hundred can be calculated.

The next section, comprised of five questions, is our main questionnaire. It aims to prove or disprove the effectiveness of the prototype’s design choices by gauging the participant’s opinion on its support for contribution transparency and traceability, the emergence of user reputation, and its potential for the expression of sociability and online deliberation. For the sake of clarity and consistency, these questions use the same five-point scale as the SUS questions that precede them.

The last section, containing just two questions, aims to provide insights into the participants’ experience in workflow modelling and Wikipedia authoring. Capturing this information makes it possible to determine whether a participant is a proficient

	Part 1 – System Usability Scale Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)
Q1	I think that I would like to use this system frequently
Q2	I found the system unnecessarily complex
Q3	I thought the system was easy to use
Q4	I think that I would need the support of a technical person to be able to use this system
Q5	I found the various functions in this system were well integrated
Q6	I thought there was too much inconsistency in this system
Q7	I would imagine that most people would learn to use this system very quickly
Q8	I found the system very cumbersome to use
Q9	I felt very confident using the system
Q10	I needed to learn a lot of things before I could get going with this system
	Part 2 – Online Collaboration Scale: 1 (Strongly Disagree) to 5 (Strongly Agree)
Q11	The prototype supports the transparency of contributions (volume, nature, importance)
Q12	The prototype supports the emergence of contributor reputation
Q13	The prototype supports the expression of sociability (etiquette and social rules)
Q14	The prototype supports the traceability of changes and decisions
Q15	The prototype supports unrestrained deliberation over workflow refinements
	Part 3 – Participant Profile Scale: 1 (Never) – 3 (Occasionally) – 5 (Regularly)
Q16	Your proficiency in workflow modelling
Q17	Your level of contribution to Wikipedia

Table 5.1: Evaluation Questionnaire

workflow modeler or a novice, which in turn helps to establish whether the prototype is acceptable to users at both ends of the spectrum—in particular, novices may be overwhelmed by too much complexity while experts may not be satisfied with a reduced feature set. The level of contribution to Wikipedia should be indicative of whether or not a given participant is familiar with the key collaboration concepts and features in Wikipedia (e.g. deliberation process, etiquette, talk pages, user pages, minor versus major edits), which may provide us with clues with regard to their evaluation. This last section also uses a five-point scale for consistency, but it is necessarily labeled differently: ratings go from *"Never"* (1) to *"Regularly"* (5), with a mid-point labeled *"Occasionally"* (3).

5.2 Pilot Study

The first two participants were scheduled two days before the remaining participants, in order to provide time to make any adjustment deemed necessary after the pilot run. Each of the pilot study participants was given an individual one-hour time slot, and their participation was closely supervised—unlike the other sessions in which participants worked mostly unobserved. These two first experiments highlighted a few minor issues, some of which affected both subjects, and some of which tended to compromise the evaluation by introducing confusion or by making relevant information more difficult to access. These issues warranted the modification of the prototype, video and instructions provided to the remaining participants. As a result, the data collected prior to these modifications is presented separately from the remaining set of experiments, and we find ourselves with two sets of data – the pilot study data of two participants, and the main study data of nine participants.

5.2.1 Pilot Study Findings

Both pilot sessions took 45 minutes, during which the participants openly discussed what they were doing and provided verbal feedback. Overall, both participants were favourably impressed with the general system, as confirmed by the results of their questionnaires.

5.2.1.1 Pilot System Usability Scale Results

According to John Brooke in his original proposal of the System Usability Scale[103], SUS questionnaire results should not be interpreted question-by-question but as a whole. We will therefore not look at a breakdown of the ten SUS questions and will focus instead on the SUS scores that are calculated from them, and on the general pattern of the SUS ratings.
The average SUS score of the pilot trial is 86%, which is a healthy score. The individual SUS scores are illustrated in figure 5.1.

When the ten SUS question results, which alternate between positively phrased questions (e.g. "I thought the system was easy to use") and negatively phrased questions (e.g. "I found the system unnecessarily complex") are plotted on a radar graph as we can see in figure 5.2, they produce an even star shape, which is also a very encouraging sign. Should there be an imbalance in the SUS ratings, it would be clearly visible on the graph; for example a consistently unfavourable score for one of the questions would produce a star shape with a misshaped branch.

The combination of these results is a clear indication that both pilot participants were overall satisfied with the usability of the prototype.

5.2.1.2 Pilot Online Collaboration Results

Figure 5.3 shows the results of the second section of the questionnaire, relating to the online collaboration features. Both users gave relatively similar marks, with two out of five questions rated identically and no more than one point difference in their rating of the remaining questions. The average rating is 4.1, with the maximum rating of 5 occurring three times, a high rating of 4 occurring five times and an average rating of 3 occurring twice. This can be interpreted as a sign of broad satisfaction with the prototype's collaboration features.
The top-rated questions were "The prototype supports the transparency of contributions (volume, nature, importance)" and "The prototype supports the traceability of changes and decisions". The lowest-rated questions were "The prototype supports the emergence of contributor reputation" and "The prototype supports the expression of sociability (etiquette and social rules)". We will see later that the follow-on study with the nine remaining participants gives a similar pattern, which we will discuss in

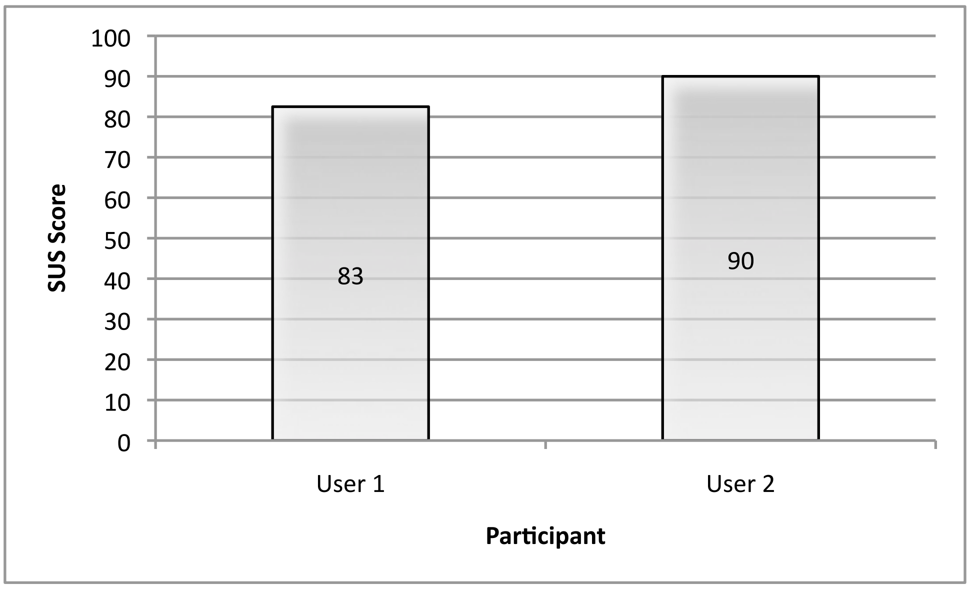


Figure 5.1: Pilot Study – SUS scores

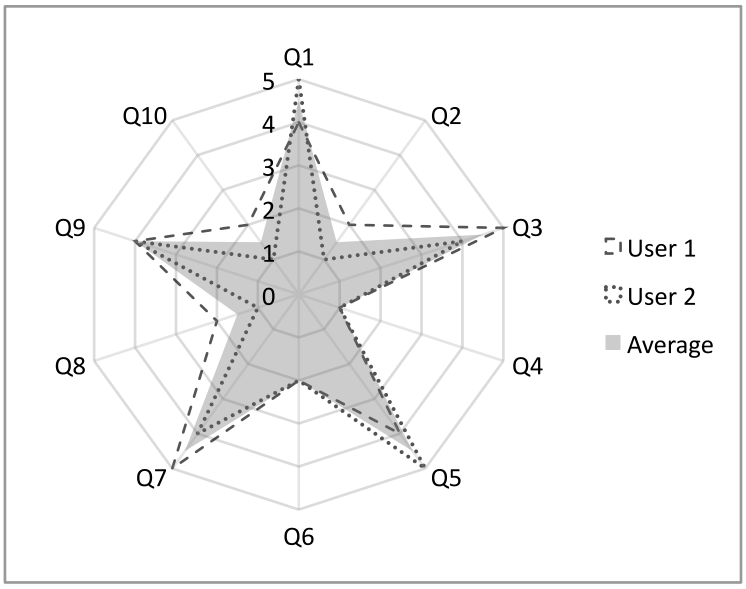


Figure 5.2: Pilot Study – SUS radar chart

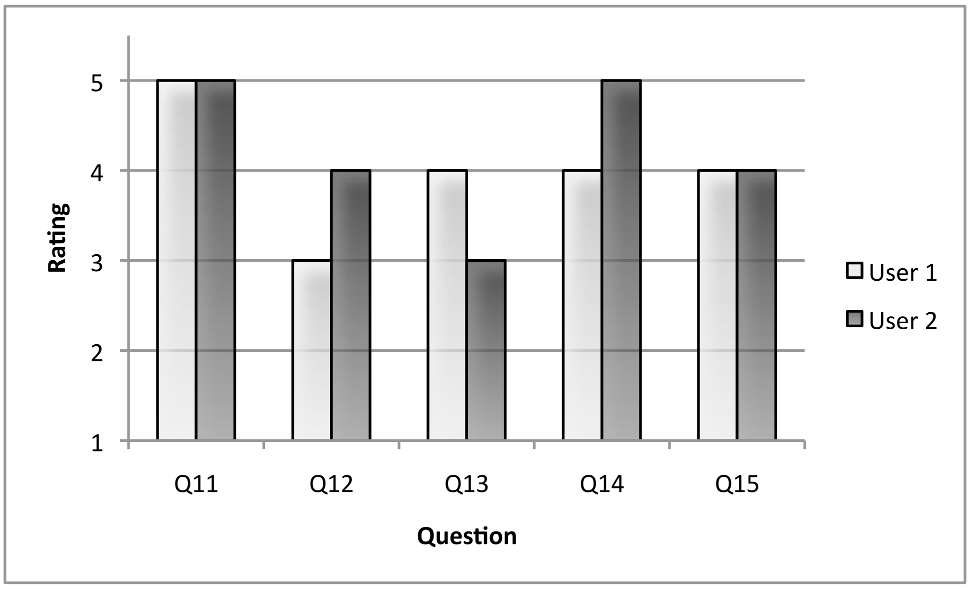


Figure 5.3: Pilot Study – Collaboration questionnaire chart

paragraph 5.4.1.

5.2.1.3 Pilot Participant Profile

Both pilot study participants were proficient workflow modelers, having given high ratings of 4 and 5 to question 16 ("Your proficiency in workflow modelling"). However neither pilot study participants were experienced Wikipedia contributors, both having given the lowest mark to question 17 ("Your level of contribution to Wikipedia"). In light of their comparable profile, the notable similarity of the ratings that they gave to the previous fifteen questions of the questionnaire gives increased confidence in the results of the pilot study—these were two comparable users who gave comparable results.

5.2.2 Pilot Comments and Observations

The pilot study participants gave the following written comments:

> Participant 1—"Very nice visualizations and intuitive to use. Some extra views would improve the tool e.g. view revision history based on user changes etc."

Participant 2—"I sometimes found it difficult to connect arcs to task boxes."

A number of minor issues were observed and verbally commented on during the pilot study:

5.2.2.1 Connecting Arcs

The way arcs were connected to shapes via drag and drop proved unintuitive. Both participants tended to move the arc until one of its ends overlapped a shape, while keeping the mouse pointer outside the bounds of this shape, expecting a connection to be made. Typically, full arcs are dragged from the toolbox by clicking roughly in their middle, and the ends of the arcs therefore remain some distance away from the mouse pointer while dragging takes place. However the application required that the mouse pointer itself must overlap a shape in order to connect the arc to it. As we have seen above, one of the participants referred to this problem in the feedback form.

5.2.2.2 Unclear Tree Control Display

The tree control shows both which node represents the "current" version, and which node is the version presently being viewed. In the pilot study, this was done using a colour-coded scheme—red for the node being viewed, green for the current version—as illustrated in figure 5.4(a). Leaving aside the accessibility problem that this represents for colour-blind users, this assumes that the user is aware of the significance of this colour scheme, and although the colour scheme was explained in the introduction video, this proved to be an unrealistic assumption. One of the two participants explicitly discussed this point and suggested that an implicit colour-coding scheme should not be relied upon.

5.2.2.3 Visual Cue For "Make Current" In Feeds

Both participants were able to locate the revision in which fictitious user Bob broke an etiquette rule in answer to exercise 2.3, but when probed about which specific action taken by Bob was rule-breaking, both were unable to give an immediate and clear answer. While pointing out the relevant feed item to them, the reason why neither participant noticed it became obvious: The action is displayed in the light-coloured heading text describing the action alongside its author and the timestamp, as illustrated in figure 5.5(a), which has all the cues of an unimportant informational line and which users would most likely not read at all. It became obvious that such an important event as marking a version current should be more prominently displayed.

5.2.2.4 Confusion With Version Numbering

In the pilot prototype, references to revision tree nodes were displayed using a dotted notation and referred to as version numbers—for example, "Version 1.2.1".
One participant was led to believe that such references carried extra significance, given their similarity with version numbers commonly used to label software releases. This became apparent when this user came to worksheet question "Identify one major revision and one minor revision (please write their version numbers)", and commented that the version numbers did not seem to match their importance. With software releases, a longer chain of dotted numbers represents a more minor release (e.g. v1.2.3 is a minor version compared to v1.2), but in our case the notation refers merely to positions in the tree and does not imply importance.

5.2.2.5 Keyboard Bug

SVG frames embedded in a web page support keyboard events, but do not capture keyboard events unless they hold keyboard focus, which typically only occurs after the user clicks on the SVG frame. One of the last features implemented before the evaluation, a minor usability improvement, consisted of detecting backspace and delete key press events at the page level and redirecting them to the SVG frame's keyboard handler, should the SVG frame not have focus. This change introduced a bug, however: When pressing delete or backspace inside a text control, such as the description field at the bottom of the page, the key press event was now redirected to the SVG frame, and in response the selected shape (if any) was deleted from the model. This late code change remained undetected until the second participant's session, where it occurred twice and caused the Start condition, which happened to be selected at that moment, to be deleted accidentally.

5.2.2.6 Other Comments and Observations

The following further issues were commented on or observed in the conduct of the pilot evaluations:

1. Presently, only one shape can be selected at a time. Adding a “group select” feature would make is easier to move a collection of shapes at once.
2. Start and End conditions can be deleted, even though they are required in YAWL workflows, and cannot be added back in as they are not featured in the toolbox. Preventing the deletion of these required shapes would be trivial to implement by displaying an alert and ignoring the delete command.

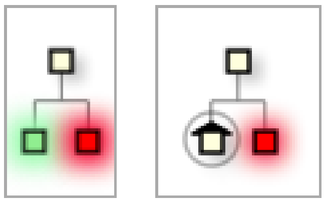
5.3 Amendments

Some of the issues that arose during the trial experiment were deemed to be potentially disruptive to the remaining evaluations and justified making amendments to the prototype. These were the lack of clarity in the encoding of information on the revision tree, the lack of visibility of the “made current” feed entries, and the potentially misleading version references. Additionally, two of the issues that arose during the use of the editor, namely the unintuitive manner of connecting arcs to shapes and the keyboard bug, were deemed to require a fix. The group select feature and the ability to delete start and end conditions were considered side issues that were unlikely to distract from the evaluation questions, and were deemed not to warrant code changes.

5.3.1 Changes Made

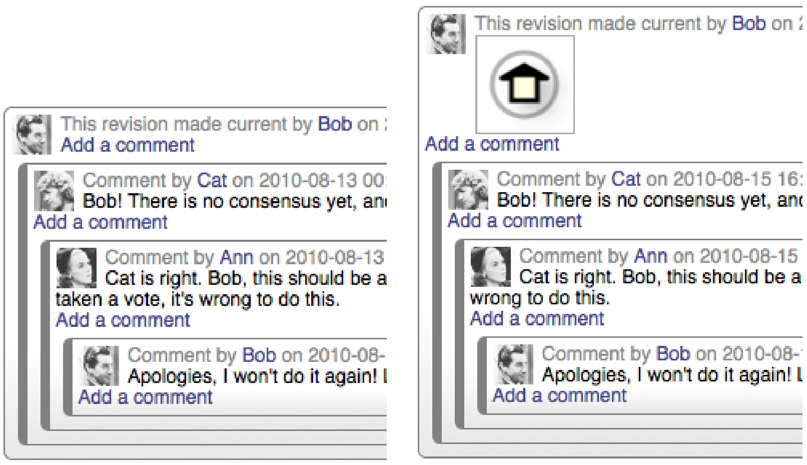
This issue with connecting arcs mentioned in section 5.2.2.1 was trivial to correct in the code. The shape connection logic is now no longer based on the mouse pointer and what shapes lies at its position (the latter being provided as an argument to the SVG *mousemove* event handler), but by the position of the arc mover controls at either end of the arc and what shape they overlap (obtained by calling SVG document object’s *elementFromPoint()* method).

To address the ambiguous tree display issue described in section 5.2.2.2, a distinguishing display feature had to be envisioned. The solution that was adopted consisted of decorating the square representing the node to give it the appearance of a home, as illustrated in figure 5.4(b). It may also be beneficial to change the terminology being used and refer to "home" rather than "current", but in the absence of supporting evidence in favour of also making this change, the terminology was left unchanged.



(a) Before (b) After

Figure 5.4: Improved visual cue in the revision tree



This revision made current by Bob on
Add a comment
Comment by Cat on 2010-08-13 00
Bob! There is no consensus yet, an
Add a comment
Comment by Ann on 2010-08-13
Cat is right. Bob, this should be a
taken a vote, it's wrong to do this.
Add a comment
Comment by Bob on 2010-08-
Apologies, I won't do it again!
Add a comment

This revision made current by Bob on
Add a comment
Comment by Cat on 2010-08-15 16:
Bob! There is no consensus yet, anc
Add a comment
Comment by Ann on 2010-08-15
Cat is right. Bob, this should be a
wrong to do this.
Add a comment
Comment by Bob on 2010-08-
Apologies, I won't do it again!
Add a comment

(a) Before (b) After

Figure 5.5: Improved display of "made current" feed entries

To resolve the visibility issue in feeds outlined in section 5.2.2.3, a visual cue was added to feed items representing making a version current. This was done by including a magnified representation of the home tree node; a particularly easy task given that it is now an SVG element that has been included in the project as per the preceding paragraph. The resulting display is shown in figure 5.5(b).

This issue detailed in section 5.2.2.4 was resolved by replacing the terminology and the notation used. References now make use of colons in place of dots and are referred to as revision nodes. For example, we now display "Revision Node 1:2:1" instead of "Version 1.2.1".

5.4 Main Study

The main study took place over two days, during which four one-hour slots were scheduled in a KDEG meeting room. To save time and reduce the use of the meeting room, participants were slotted in pairs, although they worked individually. I attended all sessions but found that participants in paired sessions were much less inclined to ask questions and to comment, given the presence of another participant in the room. The pairing also prevented me from monitoring the activity of the participants, who were seated at opposite side of the table while I sat perpendicular to them, unable to watch their screens. The only exception was participant 11, a classmate who carried out the evaluation in a different room and was not paired. Unsurprisingly, this subject made significantly more verbal comments than the paired participants and asked more questions. Given the different conditions, and the fact that participant 11, unlike the other participants, is personally known to the researcher, it was envisaged to separate his evaluation results from the remaining set or to discard it altogether. Instead, we are including this participant's data, given that the same material as the remaining set was evaluated, but pay particular attention to any divergence between this participant's data—intentionally listed last—and the remaining set.

5.4.1 Main Study Findings

5.4.1.1 System Usability Scale Results

The individual SUS scores are illustrated in figure 5.6. The average SUS score is 76.9%, with a standard deviation of 15.7%, indicative of a fairly high degree of variation. One

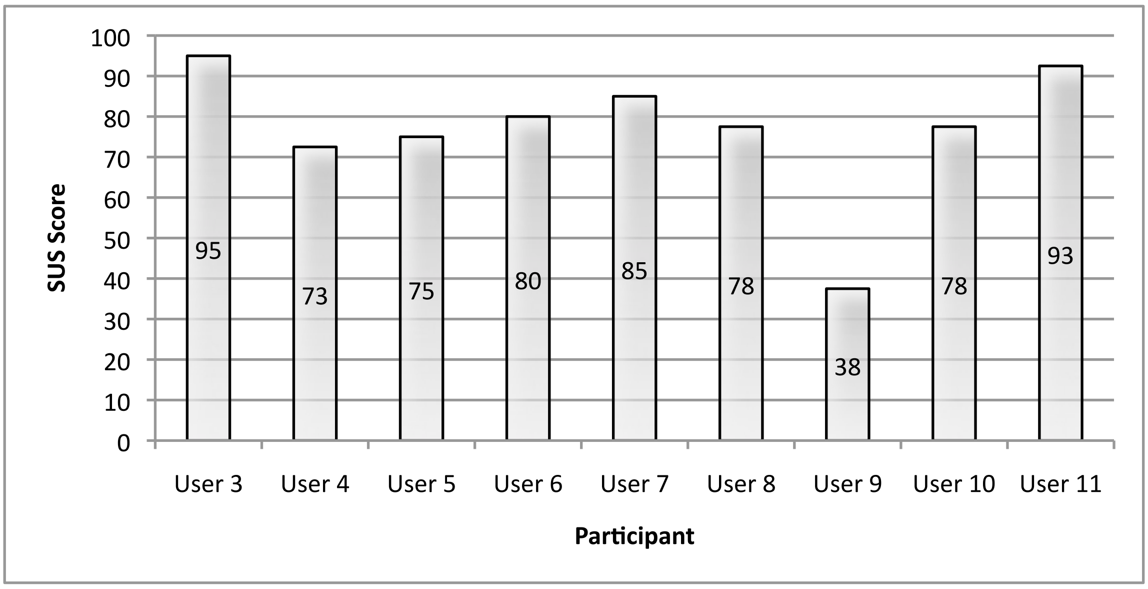


Figure 5.6: Main Study – SUS scores

score stands out as being strikingly atypical: Participant 9's SUS score is merely 38%, which is only half of the average and much lower than the second lowest score of 73%. If this participant's results were to be excluded, the standard deviation would be more than halved to a much more satisfactory 7.7% and the average would rise to 81.8%. In spite of the improvements that were made, this score remains slightly lower than the average SUS score obtained in the pilot study.

Each participant's complete SUS questionnaire results are also shown as radar charts in figure 5.7. As we can see, a number of aberrations are clearly visible in the form of stars sporting misshapen branches, but these aberrations are evenly distributed and the chart of the average scores gives a well-defined star shape, which is indicative of overall satisfaction with the system's usability. One striking exception is again participant 9's chart, which is markedly different from all the other participants' charts, even appearing as their inverse in some places.

5.4.1.2 Online Collaboration Results – Q11

Figure 5.8 shows the results for question 11: "The prototype supports the transparency of contributions (volume, nature, importance)"

With an average of 4.22 and a standard deviation of 0.79, this is the second-highest

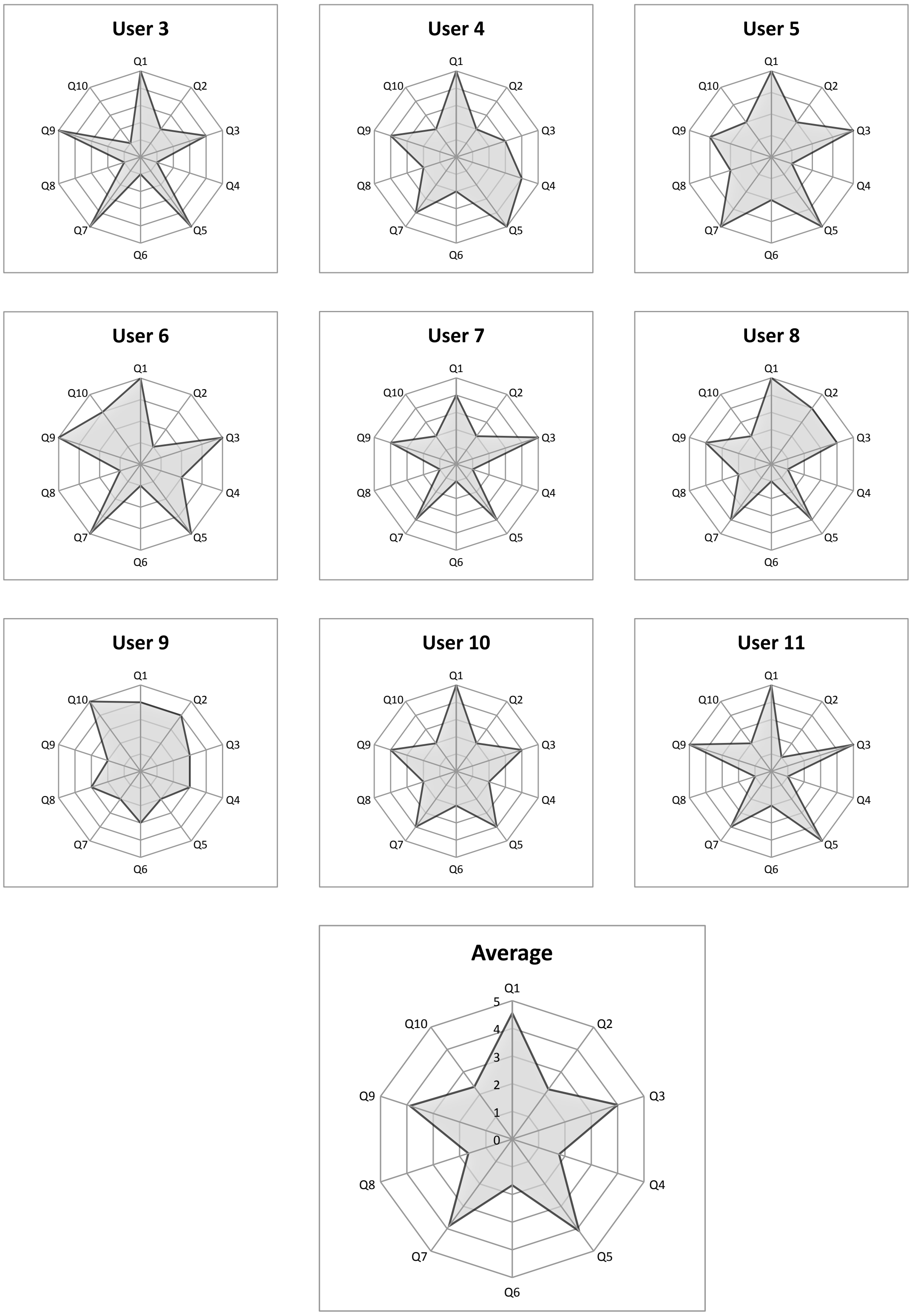


Figure 5.7: Main Study – SUS radar charts

scoring of the five online collaboration questions, with four out of the nine participants giving the top score of 5. Although in the top range for variability, this is a broad endorsement of the prototype's visibility features.

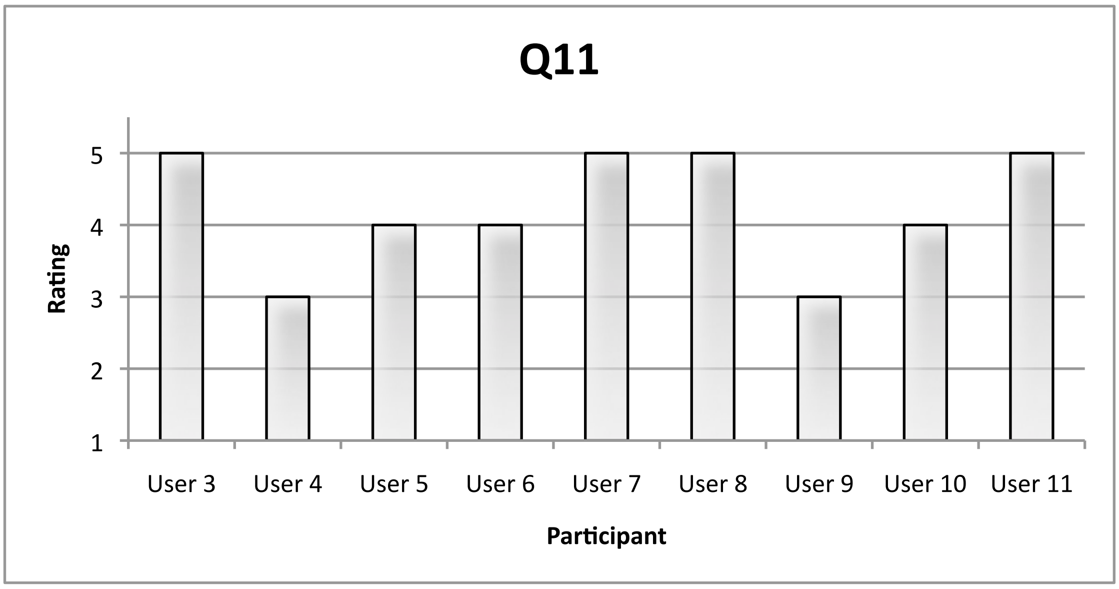


Figure 5.8: Main Study – Collaboration Q11

5.4.1.3 Online Collaboration Results – Q12

Figure 5.9 shows the results for question 12: "The prototype supports the emergence of contributor reputation"

This is the lowest scoring question, averaging 3.56 with a standard deviation of 1.07. Our atypical subject, participant 9, stands out with the minimum score of 1, "strongly disagree", a response which was not repeated by any participant in any of the remaining questions. If we take out this response, the average becomes 3.88 with a standard deviation of 0.60, a result which is obviously marginally better but also much more uniform, lending a fairly high level of confidence in the collected data in spite of the single major aberration it includes.

This question was closely related to exercise 2.1 on the worksheet: "Among Ann, Bob and Cat, who most deserves the reputation of top contributor? How did you establish this?". It was assumed that this exercise would lead the participants to use the statistics shown on the profile pages, and perhaps take a cursory look at the

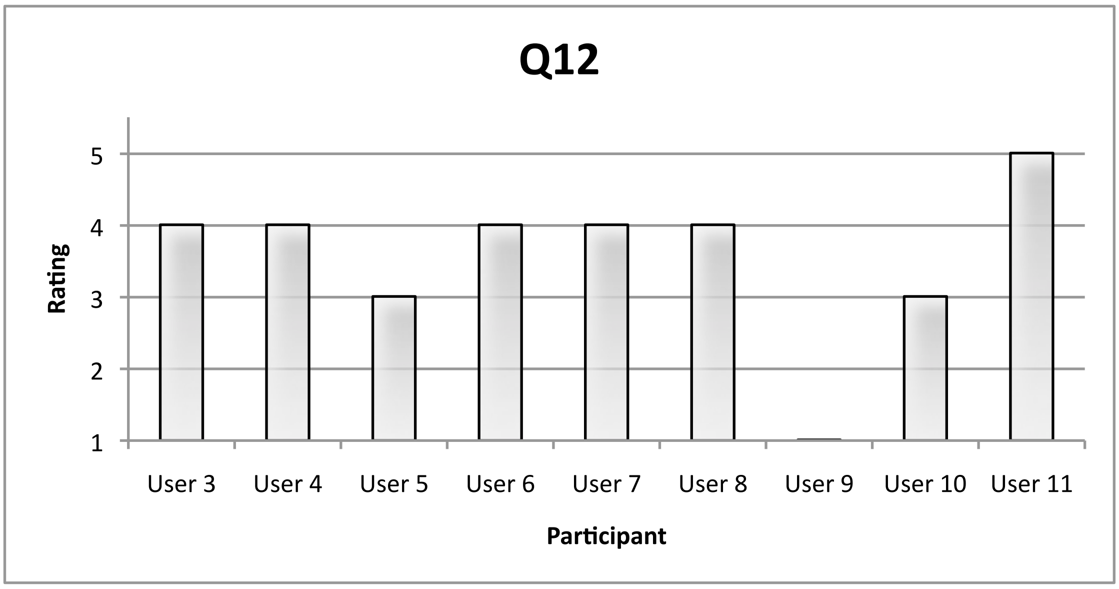


Figure 5.9: Main Study – Collaboration Q12

feeds on these three participants' profile pages, in order to contrast the relative level of contribution of fictitious users Ann, Bob and Cat and conclude that Ann deserves the best reputation. It appears from my observations that participants did not in fact tend to look at the profile pages, but rather browsed the revision tree to familiarize themselves with the contributions made. There may be a number of contributing factors:

1. The sample data set was small enough to allow participants to look at all versions, making the need for statistics less pressing than in large data sets.

2. The statistics feature was shown close to the end of the introductory video and may have failed to leave an imprint, particularly since the participants had no prior knowledge of the prototype and may not be expected to absorb the entire content of the video.

3. The low visibility of the links to profile pages (user profile pages are opened through the hyperlinked user names displayed in the heading of feed entries) may have obfuscated the only means of accessing the user statistics.

These eventualities suggest that usability and external factors such as an inadequate sample data set, rather than the general design of the prototype, could have played a part in this fairly disappointing outcome. This should ideally be verified in further evaluations.

5.4.1.4 Online Collaboration Results – Q13

Figure 5.10 shows the results for question 13: “The prototype supports the expression of sociability (etiquette and social rules)”

This is the second-lowest rating question, with an average of 3.67 and standard deviation of 0.67. Participant 9’s score no longer stands out; his rating is the lowest with a score of 3, but three other participants also share this lowest score.

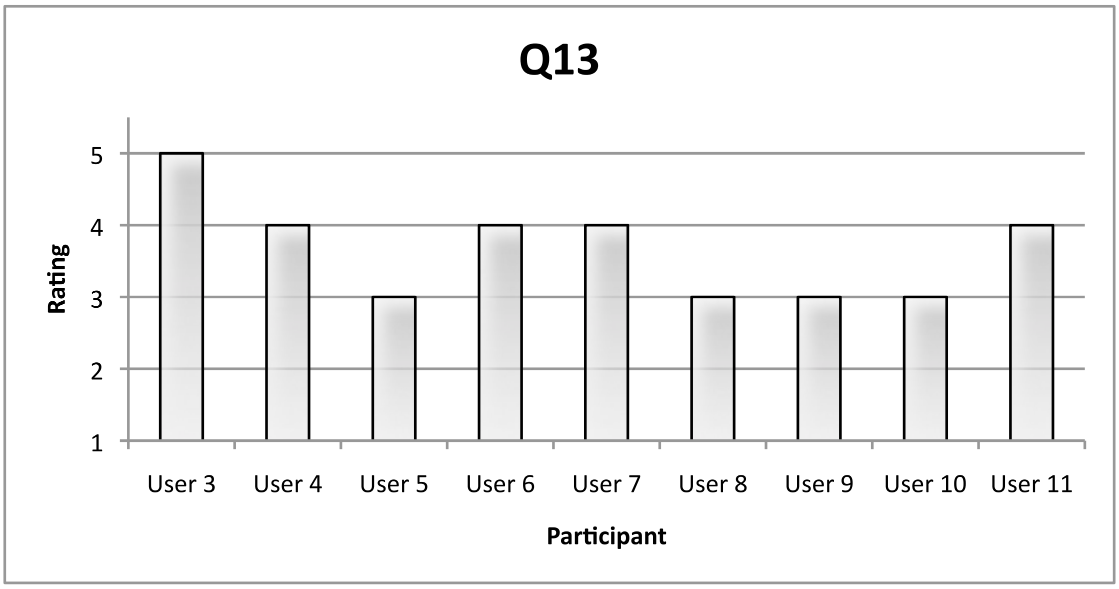


Figure 5.10: Main Study – Collaboration Q13

Interestingly, one of these lowest scorers, participant 8, gave the following written comment: ”Sociability aspect was very good and I can see how this would be useful in other applications with group/collaborative online systems.” This could be interpreted as contradictory with the average score of 3 given by this subject. Or this could support the hypothesis that the question being asked may be too vague or subject to interpretation. For instance, should the prototype be compared to social networks in answering this question?

To shed some light into what may have been the frame of reference for the participants, it may be helpful to contrast these answers with the participants’ familiarity with Wikipedia, as given in question 17. Indeed, the two participants who have indicated to have contributed to Wikipedia (although neither are experienced contributors, as we will see in section 5.4.1.7), i.e. participants 7 and 11, both gave high ratings of 4 to this question, and may have framed this question differently given their exposure to sociability in Wikipedia. It would be helpful to clarify this point in further evaluations, and to eliminate the element of doubt by rephrasing the question to make it less prone to personal interpretation.

5.4.1.5 Online Collaboration Results – Q14

Figure 5.11 shows the results for question 14: "The prototype supports the traceability of changes and decisions"

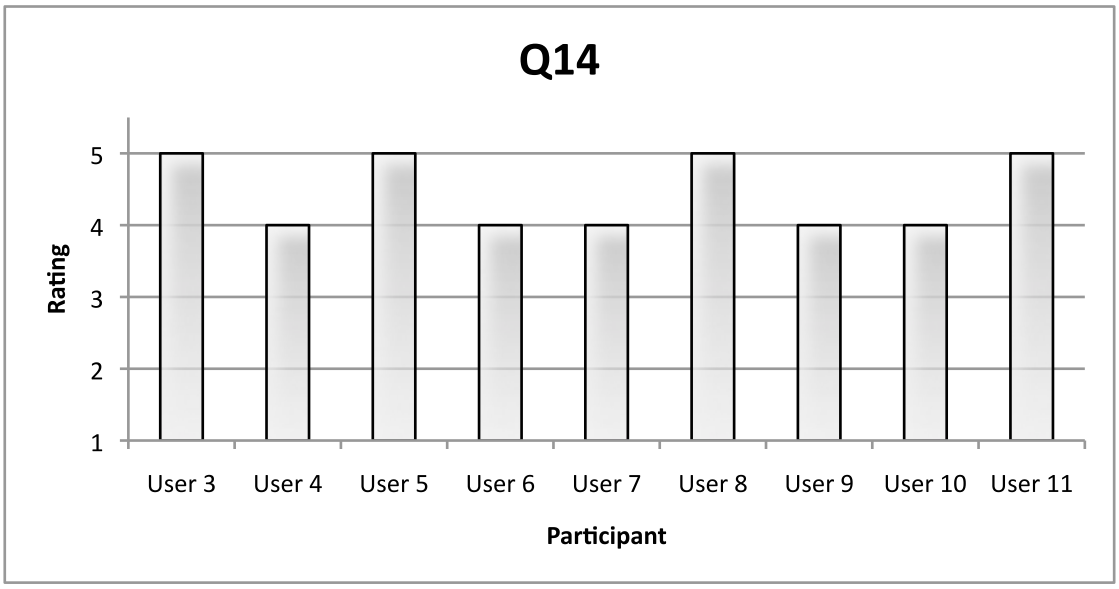


Figure 5.11: Main Study – Collaboration Q14

This question received the highest score of 4.44, with a standard deviation of 0.50. With four subjects giving this question the highest score ("Strongly agree") and the remaining five the high score of 4, our participants unanimously declare a high level of satisfaction with the prototype's performance on this point.

5.4.1.6 Online Collaboration Results – Q15

Figure 5.12 shows the results for question 15: “The prototype supports unrestrained deliberation over workflow refinements”

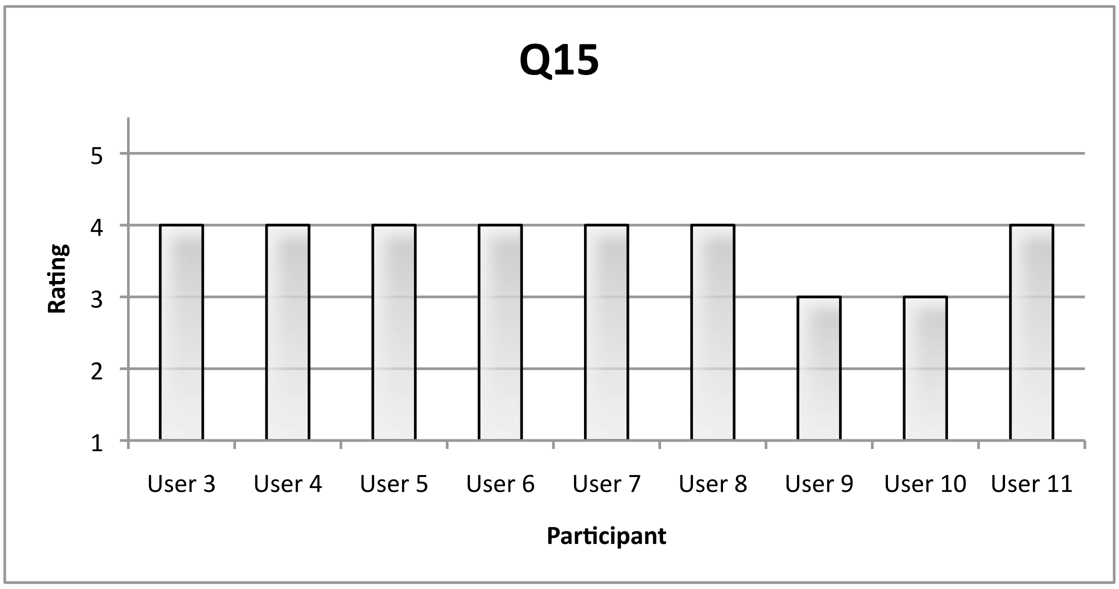


Figure 5.12: Main Study – Collaboration Q15

The average score for this question is 3.78. With a standard deviation of 0.42, the lowest of all questions, the scores are remarkably uniform. Evidently, as the scoring was done privately I did not have an opportunity to discuss scores with participants, but it would have been interesting to talk participant through was justifies a rating of 4, which was given by everyone but two participants, to ascertain what participants find to be limiting factors for deliberation. At any rate, these scores express broad satisfaction with the deliberation features.

5.4.1.7 Participant Profiles

Question 16, “Your proficiency in workflow modelling”, received six occurrences of the average score of 3 (“’occasionally”), one score of 2 and two scores of 1 (“never”). Hence these participants are significantly less skilled in workflow languages than the participants in the pilot study, but all except two do have modelling experience.
Question 17, “Your level of contribution to Wikipedia”, matched the pilot study par-

ticipants : all were absolute novices, with a score of 1 (“never”), bar two participants who gave the low score of 2. The fact that all participants bar one are involved in KDEG research means that all the issues involved are well understood, but this indicates that our subjects have little to no familiarity with the Wikipedia features which have influenced some of our design choices.

5.4.1.8 Comments

Six of the participants to the main study wrote a comment. There is no repetition or visible trend in this commenting, and none of the comments relate to the issues that we addressed following the pilot trial. This fact is a positive sign: should we have failed to resolve the most important issues which had the potential to disrupt the course of the next evaluations, we should have expected to find them mentioned in at least some of the comments.

Two of the comments provide information to help improve aspects of the prototype:

> “I would have liked it if it were possible to navigate from a sub-process to the main workflow without using the ‘back’ button”
> “The voting system was unclear (in terms of what the values really meant).”

Two comments highlighted the exaggerated simplicity of the data set and questioned what the findings would be for more complex scenarios:

> “Version branching could become quite complex fairly quickly, I wonder if the re- play/visualize could work well in comparing two versions that are far enough in the tree.”
> “I felt that the task was rather simple and possibly did not allow me to really test all the functionality of the tool.”

One participant commented on performance:

> “A few performance issues. Firefox stopped responding a few times and saving was slow.”

This is in fact unrelated to the prototype and can confidently be attributed to a general system performance issue afflicting the loaned laptop used for the evaluation, which also stopped responding while attempting to play the introduction video at the start of the evaluation session.

5.4.2 Observations

5.4.2.1 Interpretations For Participant 9

Participant 9, as we have seen, gave atypically low scores to the SUS questionnaire and to question 12. For all the other questions in the collaboration section (Q11–Q15), this participant gave a score that, while within the range of the scores given by the remaining participants, was consistently low, matching the minimum score received for each question. A clue to these disapproving results lie in the comment he gave:

> "Details buried everywhere it takes long time to find! Like the interface though."

As we have seen, our design for deliberations, with its intentional scoping across revisions and nestings, does mean that details are scattered across the data set, but visibility features should adequately counter-balance this fragmentation to make information relatively easy to find. It may be helpful to examine the exercises that may have led this participant to forming this opinion.

The worksheet included three exercises that demanded finding information in the data set. The first of these exercises, "Among Ann, Bob and Cat, who most deserves the reputation of top contributor?", may have proved frustrating for this subject assuming that he did not find the profile pages to view the user statistics. The second exercise, "Identify one major revision and one minor revision" could have been time-consuming if taken literally, as the application makes no mention of major and minor version, but highlights structural, textual, cosmetic and linking revisions, from which importance was expected to be interpreted. Indeed, another participant may have been led astray by this fact, as shown by his own comment:

> "I didn't notice an explicit note making a version major, if there isn't (maybe I missed it), I think there should be."

The third exercise, "Find one instance of bad behaviour by Bob" may again have proved particularly frustrating if the participant could not find the user page for Bob, which contains this user's feed. Additionally, the introductory video failed to mention the expandable feed of all revisions of a model at the bottom of the view pages, which would have simplified searching for information across revisions.

All the factors above may have influenced the comment given by this user, which is our best clue for understanding the low ratings that he gave in the questionnaire.

5.4.2.2 Issues Not Noticed In The Pilot

One participant asked about the ordering of items in the feeds. The feeds list the newest entries at the top, however the video inaccurately states that feeds are displayed "chronological order" where it should have been "reverse chronological". It became apparent that adopting this ordering is not only counter-intuitive for some of the participants, but also introduces inconsistency. Responses to comments, which are displayed as nested frames below the original comment, inherently follow a chronological ordering. Hence feeds may contain a mixture of reverse chronological and chronological ordering. It would appear that chronological ordering would have been preferable for the sake of consistency and clarity.

5.4.2.3 Influence Of Trial Conditions

It is quite striking that in spite of the improvements made to the material, the results of the main trial are marginally lower than the results of the pilot trial. One possibility is that the conditions of the evaluation, and more specifically the significantly higher level of interpersonal communication experienced in the single user sessions in contrast with the paired sessions, may have played a part. The results of participant 11, which ranks second highest in the set, to some degree support this hypothesis since this was the only participant of the main study who was not paired; although the fact that this was a classmate was likely to have also played a part. The atypical results of participant 9, which as we have discussed may stem from a failure to find the relevant pages, might also have been different had this participant been given an individual session, which may have encouraged him to ask relevant questions. Further trials may help refute or confirm the suggestion that the pairing of participants negatively affected the results.

5.5 Evaluation Summary

A video and an evaluation worksheet comprising an exercise and a series of questions were produced to expose the participants to the key features of the prototype with regard to the research: visibility, reputation, sociability, transparency, deliberation and ease of use.
Eleven participants with no prior knowledge of the research were surveyed after using the prototype while completing the instructions in the worksheet. The first two participants were part of a pilot study and were monitored individually, allowing for more verbal feedback. The remaining participants worked independently, but could ask questions.
The data shows encouraging average results for system usability and online collaboration. The lowest-scoring questions concern the ability of the system with regard to the emergence of reputation and sociability. The limited, fictitious data set and the short exposure to the application and its data may be contributing factors. The remaining questions support the view that the prototype adequately supports deliberation and provides for transparency and traceability.
Further evaluations would help address some of the new questions raised by our findings.

Chapter 6

Future Work

6.1 Data Flow Modelling

As we explained in section 3.2.1, we chose to focus on non-executable workflows and not concern ourselves with the data perspective. This confines the applicability of the research to one end of the spectrum of application domains (as detailed in section 2.2.2.3) and leaves execution-driven scenarios out of reach. While such scenarios tend to occur in the realm of industry and in the presence of formal hierarchy, there is undeniably a case for workflow-driven execution to support the more sophisticated cases of online collective action. Consequently, this research should be extended to contexts of collective modelling in which complete and correct data definitions are required.

6.2 Further Evaluations

Practical aspects have constrained our evaluation to a relatively short period of time. It was therefore not possible to evaluate our solution in the context of non-simulated multi-participant collaboration, nor were we able to collect data on sustained, repeated usage of the prototype. Further evaluations, in particular a longitudinal study, would help to assess the validity of our findings and draw further conclusions.

6.3 Applicability to Other Domains

It would be interesting to review other collective modelling domains to establish whether our findings apply elsewhere. For instance UML diagrams are graphical models comparable with workflows (indeed, we saw in section 2.2.2.1 that UML Activity Diagrams can be used as workflows), and UML modelling generally occurs in teams in a context of collective ownership. Additionally, open source software development communities engage in a remarkable form of online collective action that may benefit from a suitable approach to collaborative UML diagram editing. The emergence of social collaboration networks for programmers such as GitHub[104] may provide a platform for experimentation.

6.4 Real-World Applicability

The Centre for Next Generation Localisation (CNGL), a research centre funded by the Irish government and spanning four universities and a number of industrial partners, has initiated a review of this research and the prototype we developed. This group has expressed an interest in reusing our code as part of an online collaboration system managing the collective authoring of content process flows.

Chapter 7

Conclusions

Through the completion of this research, including the design and evaluation of a prototype, we have found evidence that workflow modelling is well suited to online collaboration scenarios:

1. Workflows are inherently structured in comparison with other types of documents such as free text, and this structure can be utilized to facilitate change management. For example, while wiki software used to collectively edit natural language documents cannot distinguish major and minor edits without soliciting user input, workflow applications can automatically and reliably categorize changes made to a workflow, as demonstrated by our prototype. These can then be used to support effective visualizations.

2. Multi-track deliberation on workflow modelling appears to be intuitive and usable, according to the evaluation results of our prototype, which supports it through a tree structure of proposed revisions. Nestings, when supported by the workflow language (e.g. YAWL and BPMN), provide a further opportunity for the intuitive containment of deliberations.

3. On a more practical note, although it is an important consideration for supporting a diverse online community, we have shown that Web standards such as SVG, CSS and JavaScript provide a workable basis for the implementation of a Web-based workflow modelling application.

Further empirical studies may help to confirm whether these findings can be exploited successfully in real-world self-managed environments, and whether online communities can apply collaborative workflow modelling to support their conduct of online collective action.

Appendix A

Abbreviations

Short Term	**Expanded Term**
AJAX	Asynchronous JavaScript and XML
BPEL	Business Process Execution Language
BPM	Business Process Modeling – or – Business Process Management
BPMS	Business Process Management System
CMC	Computer Mediated Communication
CMS	Content Management System
CSS	Cascading Style Sheets
DOM	Document Object Model
GUID	Globally Unique Identifier
JSON	JavaScript Object Notation
KDEG	Knowledge and Data Engineering Group
OASIS	Organization for the Advancement of Structured Information Standards
ODE	Orchestration Director Engine (as in Apache ODE)
OMG	Object Management Group
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
PII	Personally Identifiable Information
RIA	Rich Internet Application
SUS	System Usability Scale
SVG	Scalable Vector Graphics
UI	User Interface
UML	Unified Modeling Language
W3C	World Wide Web Consortium
XSS	Cross-Site Scripting
YAWL	Yet Another Workflow Language

Appendix B

Evaluation Video Transcript

“Welcome to this quick overview.
First we will create a new model to look at the editor. Modeling is done in the YAWL workflow language. In YAWL, every workflow includes a start condition, an end condition, and any number of the shapes shown on the left.
You can drag and drop shapes onto the canvas to add them to your model or move existing shapes. Double-clicking an item lets you edit its text. In this case I am modeling my evening routine and feeding the cats is my first task.
You can add sub-processes to your model by dragging nestings onto the canvas. Once you do, controls are added to let you choose whether to create a new model or to pick an existing model for your sub-process.
Arcs can be dragged onto shapes to connect them, after which they become dependent on them. You can align shapes by holding down the shift key, which snaps them to a grid.
The decorators on the left are YAWL syntactic constructs which you can drag onto tasks. These are joins and splits, this one represents multiple tasks and the frame is a cancellation area.
You can resize items by selecting them and dragging their resizers, and you can delete items by pressing the delete key on your keyboard.
Hit ‘Save’ and your first model has been created.

Let’s go back to the home page to look at the collaboration features. We’ll take an

existing workflow. This is a workflow view page. You can see that this workflow has a nesting which is shown on the right; you can click on it to open it.
Every nesting is a separate document and has its own version tree.
The version tree is shown here on the left. The revision number given at the top matches the position in red on the tree. You can click on different nodes to open them. The system maintains a full tree of revisions in order to facilitate tracking and transparency, and also to allow for deliberations to take place in different branches of the tree.
The frame at the top is a notification frame. You may see this in various versions to inform you of the particular status of the document.
This one is a dead branch, which means that since it has been created it has been abandoned and that no further work should be done from it.
The actions you can take are Edit and Make current. The current version of the workflow is shown as the home icon on the tree. There is always one, and only one, current version of a workflow. While work and deliberations may take place anywhere, a version would typically only be marked current once the contributors are satisfied that it is the one that is most ready for publication and public consumption.
You can mark a revision as dead as we have seen, but you can also mark it as draft to inform collaborators that it is still work in progress. And you can give ratings to any revision.
And below we can see the changes that were made. When a model is saved, users can type a description, but it cannot always be trusted. The tool however keeps track of the changes that were made and gives you a digested view of these changes. In this case we can see that this was a cosmetic revision. You can visualize the changes on the model. You can see here in red that this decorator has been added. Indeed, this is what is listed here. You can also replay the changes.
And there is a color-coded scheme for changes, which makes them more apparent. Let's have a look at another version. This one also has cosmetic and textual changes as well as structural.
Below this is the feed for this version. Everything that has been done to this version is shown here, including what we just did when we rated the model and marked it as current and draft. You can also see comments that users may have entered on each other's entries and we can add our own comment, let's say for example "I agree".

You can also look at user pages. If you click the name of a user you can see their profile page where their statistics are displayed. We can see how many top and nested models this user has created, how many revisions were made in total, how many are draft and how became dead branches, how many comments were posted and how many votes were made, and what's the average.
You can also see a feed of all the contributions the user has made, in chronological order.
Finally, we can compare not only the changes that were made between one version and its preceding version, but across the tree. For example if we want to see the differences between this node here and the current node we can click compare and choose this node, which means going up the tree and back down. And we are presented again with the controls we have seen before, where you can see a list of the changes and visualize them on the model, or indeed animate them.
This concludes our introduction. Thank you for watching."

Appendix C

Evaluation Worksheet

Prerequisite: Please watch the video, which introduces the necessary concepts and features.

C.1 Exercise: Modeling/Versioning Warm-Up

C.1.1 Initial model (3 min)

Setup: From the home page, click "Create a New Model"

You are asked to model a typical morning routine. It includes three sequential tasks: task *Wake*, task *Make Coffee* and task *Shower*.
Create your model by adding tasks and connecting them with arcs. The accuracy of your models is of no significance to this study and you are only expected to use rectangles and arcs, but you may try more shapes if you wish. Save your workflow as "Morning Routine".

C.1.2 First two branches (5 min)

You wish to include listening to the radio as part of the morning routine. Task *Radio* can occur after Wake (sequentially) or at the same time as Make Coffee (in parallel). You are asked to create a new version for both, based on the model you created in 1.1.

C.1.2.1 First branch

Setup: From the current page (revision node 1), click "Edit'
Add task *Radio* between Wake and Make Coffee, reconnect the model using arcs, and click Save.

C.1.2.2 Second branch

Setup: Click the top node in the tree to leave revision node 1:1 and return to revision node 1. Then click "Edit'
Add task *Radio* above Make Coffee, add extra arcs to connect it to Wake and Shower, and click Save.

C.2 Exercise: Interpreting Sample Activity

Setup: In the top bar, click Home, then open the "Order Fulfilment" model. You may now navigate through any part of the system to answer each question.

C.2.1 Question (5 min)

Among Ann, Bob and Cat, who most deserves the reputation of top contributor? How did you establish this?

C.2.2 Question (3 min)

Identify one major revision and one minor revision (please write their version numbers). How could you tell that the major revision you identified is more important than the minor one?

C.2.3 Question (5 min)

In online communities, a code of practice often evolves to form guidelines on what should and should not be done. Find one instance of bad behaviour by Bob. How did you find it? How did the online community address it?

Appendix D

Data Collected

System Usability Scale Questions

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
Pilot Study											
User 1	4	2	5	1	4	2	5	2	4	2	82.5
User 2	5	1	4	1	5	2	4	1	4	1	90
Main Study											
User 3	5	2	4	1	5	1	5	1	5	1	95
User 4	5	2	3	4	5	2	4	2	4	2	72.5
User 5	4	2	4	1	4	2	4	2	3	2	75
User 6	4	1	4	2	4	1	4	1	4	3	80
User 7	4	2	5	1	4	1	4	1	4	2	85
User 8	5	4	4	1	4	1	4	2	4	2	77.5
User 9	4	4	3	3	2	3	2	3	2	5	37.5
User 10	5	2	4	2	4	2	4	2	4	2	77.5
User 11	5	1	5	1	5	2	4	1	5	2	92.5

Table D.1: Questionnaire Results – Questions 1 to 10

Online Collaboration Questions

	Q11	**Q12**	**Q13**	**Q14**	**Q15**	**Average**
Pilot Study						
User 1	5	3	4	4	4	4
User 2	5	4	3	5	4	4.2
Main Study						
User 3	5	4	5	5	4	4.6
User 4	3	4	4	4	4	3.8
User 5	4	3	3	5	4	3.8
User 6	4	4	4	4	4	4
User 7	5	4	4	4	4	4.2
User 8	5	4	3	5	4	4.2
User 9	3	1	3	4	3	2.8
User 10	4	3	3	4	3	3.4
User 11	5	5	4	5	4	4.6

Table D.2: Questionnaire Results – Questions 11 to 15

Participant Profile Questions and Comments

	Q16	Q17	**Comments**
Pilot Study			
User 1	4	1	Very nice visualizations and intuitive to use. Some extra views would improve the tool e.g. view revision history based on user changes etc.
User 2	5	1	I sometimes found it difficult to connect arcs to task boxes.
Main Study			
User 3	3	1	I would have liked it if it were possible to navigate from a sub-process to the main workflow without using the “back” button
User 4	1	1	
User 5	1	1	A few performance issues. Firefox stopped responding a few times and saving was slow.
User 6	3	1	I thought the solution was clear and well laid out given the complexity of the problem you are trying to solve (a group workflow modelling system)
User 7	3	2	Users don’t seem to have absolute rating (you can see their contribution but a lot of contribution does not indicate quality), maybe this is covered by the rating of revision but maybe it should be separate. Version branching could become quite complex fairly quickly, I wonder if the replay/visualize could work well in comparing two versions that are far enough in the tree. I didn’t notice an explicit note making a version major, if there isn’t (maybe I missed it), I think there should be.
User 8	2	1	I felt that the task was rather simple and possibly did not allow me to really test all the functionality of the tool. The voting system was unclear (in terms of what the values really meant). User interface was effective and user friendly. Sociability aspect was very good and I can see how this would be useful in other applications with group/collaborative online systems.
User 9	3	1	Details buried everywhere it takes long time to find! Like the interface though.
User 10	3	1	
User 11	3	2	

Table D.3: Questionnaire Results – Questions 16, 17 and Comments

Appendix E

Security Aspects

E.1 Overview

This appendix is adapted from coursework completed as part of a security module, for which students were asked to write about the security aspects of their dissertation project.

While it is tempting to dismiss security issues as orthogonal to a research project, the subject of online community is pervaded by security and privacy concerns. This is made clear in *Online Communities: Designing Usability and Supporting Sociability*[18]:

> "The success of many—maybe most—online communities will be strongly influenced by how secure they are. Personal information of all kinds must be secure, which means not only that systems be made secure, but that users *perceive* them as secure."

Proposed Future work suggests evaluating the prototype "in the wild" to assess its sustained performance in an online community. This likely assumes taking the prototype out of its controlled lab environment and making it widely accessible on the Web, thus exposing it to a much more challenging security context. Securing the application will require addressing generic web development, Internet and open source security issues, and issues specific to the application domain.

E.2 Generic Web Application Issues

Web Applications are by nature particularly exposed to attacks. We can look at the latest OWASP (Open Web Application Security Project)[105] guidelines for typical web application vulnerabilities and their corresponding mitigation strategies. We will detail just the top three in this chapter:

E.2.1 Cross Site Scripting (XSS)

XSS attacks could compromise our user authentication system by allowing an attacker to craft malicious input causing client-side code to access a legitimate user's cookie. A number of mitigation techniques exist, but the key strategy consists of not trusting user input and rejecting any input that does not pass strict validation.

E.2.2 Injection Flaws

Our CMS relies on a database to store all its data. SQL injection attacks could compromise all this data – reading, modifying or deleting it. Mitigations include strict input validation (as above), using secure SQL APIs (e.g. using query parameters rather than building SQL query strings on the fly) and ensuring that the database account being used by the web application is configured with the minimum sufficient set of privileges (to preventing malicious requests such as "drop table" from executing).

E.2.3 Malicious File Execution

Although not a planned feature, we may envisage enabling filename input or file uploads, for example to allow users to import workflows into the system rather than design them from scratch. Should we do so, we need to pay close attention to malicious file execution vulnerabilities. Our server side components are written in PHP, which is particularly vulnerable to Remote File Include (RFI) attacks[105]. As attacks typically rely on malicious file names, a mitigation strategy consists of not allowing user input to influence file names used in any server-side file operation, which can for example be achieved by generating unique filenames on the fly on the server side rather than allow user input to determine file names on the server.

E.3 Generic Internet Issues

Beyond Web developement issues, there are also generic server and network infrastructure-related vulnerabilities, which include Denial of Service (DoS) attacks, worms, and eavesdropping. Security must play an essential part in our deployment planning. For example, the database server should not be accessible from the Internet and should be behind a firewall, and network events should be logged to detect suspicious activity and respond to attacks. Also, given our relatively low expected traffic volumes, it is reasonable to use the TLS protocol for all exchanges, including AJAX requests.

E.4 Specific Issues

The nature of the application itself introduces specific vulnerabilities. Workflows can represent sensitive information, which has repercussions on our threat model. Also, mischievous edits are a particular concern for online collaboration frameworks.

E.4.1 Threat Modeling

Of the STRIDE[106] threats, our application is particularly vulnerable to Information Disclosure, Tampering, Spoofing and Repudiation threats:

E.4.1.1 Information Disclosure

Workflows may be of a strategic nature, and failing to keep them secret may be dam-ageable to the organization. For example, the system may be used to design a workflow representing physical security procedures in a company (such as the process for issuing employees with their secure access badges) and knowledge of this workflow can help attackers plan an intrusion.

E.4.1.2 Tampering with Data

Beyond gaining access to sensitive information to discover it, attackers could also intro-duce malicious changes to sensitive workflows. In particular, an attack strategy could be built around the ability to modify a business process in a subtle way in order to introduce a breach that can be exploited later. This could easily be made more difficult

to detect by impersonating a legitimate user.
For example, an attacker could make a subtle change to the business process driving the attribution of secure access badges, introducing a carefully crafted loophole in the process that permits them to illegitimately obtain access.

E.4.1.3 Spoofing

The previous two paragraphs leads to the importance of enforcing strong authentication security. We have seen the risk XSS causes by compromising user session cookies, which can be replayed by the attacker to authenticate as a valid user.
Also, it is important to adopt a strong password policy to guard against dictionary and brute force password attacks and to request that passwords be reset on a regular basis.

E.4.1.4 Repudiation

To deal with vandalism, voting fraud and attacks, we will require the ability to trace client activity. It is therefore important to log all relevant information such as IP addresses and timestamps.

E.4.2 Online Collaboration Issues

E.4.2.1 Vandalism

Wikipedia is a good example of a collaborative platform whose openness makes it prone to mischievous edits. Vandalism can be addressed as a security issue or as a governance and monitoring issue. The latter is Wikipedia's choice – most articles can be edited by anyone, even anonymously, and Wikipedia's effectiveness in handling vandalism is due to its the efficient and cheap monitoring features, in particular the watch lists[9], combined with its large user base. In our case, monitoring may not be as cheap: vandalism in a workflow is less evident than in text, easier to hide (particularly in complex workflows, which can include nested sub-processes) and takes more time to notice. Therefore we will need security measures to counter vandalism, such as requiring verified accounts for authentication.
Related to vandalism is the issue of self-interested edits, for example by lobby groups,

which are more difficult to detect and are particularly problematic for Wikipedia[14]. In our case, given the more cohesive nature of our users, this is not likely to be a problem.

E.4.2.2 Voting

Our decentralized authority user model relies on making decisions through consensus-building, which will include an element of voting. To mitigate fraud such as multiple votes per user, we can rely on verified accounts. We can also make use of our community of users to notice and report irregular patterns by making the lists of voters visible.

E.4.2.3 Verified credentials

We have established the need for verified credentials with regard to vandalism and voting. We may verify an account at registration time by requesting and verifying a valid email address within a restricted set of domains (for example email addresses ending with tcd.ie). Verification can be achieved by emailing a secret to the email address provided, and requesting that the user provides this secret in order to complete registration.
Adding verification of this kind, however, introduces the issue of Personally Identifiable Information (PII), which must be secured and whose storage may be subject to user consent.

Bibliography

[1] T. O'Reilly, "What is Web 2.0," *Design patterns and business models for the next generation of software*, vol. 30, p. 2005, 2005.

[2] T. Lee, "The world wide web: Past, present and future," *IEEE Computer special issue of October*, vol. 1996, 1996.

[3] T. Malone, "Is 'empowerment' just a fad? Control, decision-making, and information technology," *BT Technology Journal*, vol. 17, no. 4, pp. 141–144, 1999.

[4] J. Surowiecki, *The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations.* Doubleday Books, 2004.

[5] M. Klein, C. Dellarocas, and A. Bernstein, "Introduction to the special issue on adaptive workflow systems," *Computer Supported Cooperative Work (CSCW)*, vol. 9, no. 3, pp. 265–267, 2000.

[6] C. Shirky, *Here comes everybody: the power of organizing without organizations.* Penguin Pr, 2008.

[7] E. Ostrom, *Governing the commons: The evolution of institutions for collective action.* Cambridge Univ Pr, 1990.

[8] G. Hardin, "The tragedy of the commons," *Science*, vol. 162, no. 3859, pp. 1243–1248, 1968.

[9] F. Viégas, M. Wattenberg, and M. McKeon, "The hidden order of Wikipedia," *Lecture notes in computer science*, vol. 4564, p. 445, 2007.

[10] I. Jahnke, "Socio-technical communities: From informal to formal," *Handbook of Research on Socio-Technical Design and Social Networking Systems. IGI Global Publisher*, pp. 763–778, 2009.

[11] D. Lewis, J. McAuley, and K. Feeney, "A platform for studying progressive self management in online communities," in *Web Science Conference, Athens, Greece, 18-20 March 2009*, 2009.

[12] F. Viegas, M. Wattenberg, J. Kriss, and F. Van Ham, "Talk before you type: Coordination in Wikipedia," in *40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007*, pp. 78–78, 2007.

[13] T. Hassine, "The dynamics of NPOV disputes," in *Proceedings of Wikimania*, 2005.

[14] J. Borland, "See who's editing Wikipedia—Diebold, the CIA, a campaign," *Wired Digital*, vol. 14, 2007.

[15] A. Kittur, B. Suh, B. Pendleton, and E. Chi, "He says, she says: Conflict and coordination in Wikipedia," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 462, ACM, 2007.

[16] D. Riehle, "How and why Wikipedia works: an interview with Angela Beesley, Elisabeth Bauer, and Kizu Naoko," in *Proceedings of the 2006 international symposium on Wikis*, p. 8, ACM, 2006.

[17] A. De Moor and M. Aakhus, "Argumentation support: From technologies to tools," *COMMUNICATIONS-ACM*, vol. 49, no. 3, p. 93, 2006.

[18] J. Preece, *Online Communities: Designing Usability and Supporting Sociability.* John Wiley & Sons, Inc. New York, NY, USA, 2000.

[19] C. De Souza and J. Preece, "A framework for analyzing and understanding online communities," *Interacting with Computers*, vol. 16, no. 3, pp. 579–610, 2004.

[20] P. Lowry, T. Roberts, N. Romano Jr, P. Cheney, and R. Hightower, "The impact of group size and social presence on small-group communication: Does computer-

mediated communication make a difference?," *Small Group Research*, vol. 37, no. 6, p. 631, 2006.

[21] A. Mehrabian, *Silent messages: Implicit communication of emotions and attitudes.* Wadsworth Pub Co, 1981.

[22] J. Walther and K. D'Addario, "The impacts of emoticons on message interpretation in computer-mediated communication," *Social Science Computer Review*, vol. 19, no. 3, p. 324, 2001.

[23] G. Decker, H. Overdick, and M. Weske, "Oryx—An Open Modeling Platform for the BPM Community," in *BPM*, vol. 5240, pp. 382–385, Springer.

[24] "Oryx Developer Network: Oryx delivers process sketches." http://bpt.hpi.uni-potsdam.de/Oryx/DeveloperNetwork, last accessed 7/9/2010.

[25] "Wikipedia help: Minor edit." http://en.wikipedia.org/wiki/Help:Minor_edit/, last accessed 3/09/2010.

[26] T. Ellkvist, D. Koop, E. Anderson, J. Freire, and C. Silva, "Using provenance to support real-time collaborative design of workflows," in *Provenance and Annotation of Data and Processes*, p. 279, Springer, 2008.

[27] F. B. Viégas, M. Wattenberg, and K. Dave, "Studying cooperation and conflict between authors with history flow visualizations," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, (New York, NY, USA), pp. 575–582, ACM, 2004.

[28] A. Grosskopf, J. Brunnert, S. Wehrmeyer, and M. Weske, "BPMNCommunity.org: A Forum for Process Modeling Practitioners–A Data Repository for Empirical BPM Research,"

[29] T. Hampel, R. Keil-Slawik, B. Claassen, F. Plohmann, and C. Reimann, "Pragmatic solutions for better integration of the visually impaired in virtual communities," in *Proceedings of the international ACM SIGGROUP conference on Supporting group work*, pp. 258–266, ACM New York, NY, USA, 1999.

[30] F. Taylor, "Scientific management," *New York*, 1911.

[31] H. Braverman, “The degradation of work in the twentieth century,” *The sociology of organizations: Classic, contemporary, and critical readings*, pp. 32–38, 2003.

[32] A. Maslow, “A theory of human motivation,” *Psychological review*, vol. 50, no. 4, pp. 370–396, 1943.

[33] P. Bain, A. Watson, G. Mulvey, P. Taylor, and G. Gall, “Taylorism, targets and the pursuit of quantity and quality by call centre management,” *New Technology Work and Employment*, vol. 17, no. 3, pp. 170–185, 2002.

[34] R. Karasek, “Lower health risk with increased job control among white collar workers,” *Journal of Organizational Behavior*, vol. 11, no. 3, pp. 171–185, 1990.

[35] O. Marjanovic, “Supporting the “soft” side of business process reengineering,” *Business Process Management Journal*, vol. 6, no. 1, pp. 43–53, 2000.

[36] D. Knights and D. McCabe, “What happens when the phone goes wild?: Staff, stress and spaces for escape in a bpr telephone banking work regime,” *Journal of Management Studies*, vol. 35, no. 2, pp. 163–194, 2002.

[37] D. Yeatts and C. Hyten, *High-performing self-managed work teams: A comparison of theory to practice.* Sage Pubns, 1998.

[38] D. Georgakopoulos, M. Hornick, and A. Sheth, “An overview of workflow management: From process modeling to workflow automation infrastructure,” *Distributed and parallel databases*, vol. 3, no. 2, pp. 119–153, 1995.

[39] M. Fowler and K. Scott, *UML distilled: a brief guide to the standard object modeling language.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2000.

[40] M. Dumas and A. ter Hofstede, “UML activity diagrams as a workflow specification language,” *UML 2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools*, pp. 76–90, 2001.

[41] “Workflow Management Coalition (WfMC).” http://www.wfmc.org/, last accessed 9/04/2010.

[42] C. Ouyang, M. Dumas, S. Breutel, and A. ter Hofstede, "Translating standard process models to bpel," in *Advanced Information Systems Engineering*, pp. 417–432, Springer, 2006.

[43] C. Ouyang, W. Van Der Aalst, M. Dumas, and A. Ter Hofstede, "Translating BPMN to BPEL," *BPM Center Report BPM-06-02, BPMcenter. org*, 2006.

[44] W. Van Der Aalst, A. Ter Hofstede, B. Kiepuszewski, and A. Barros, "Workflow patterns," *Distributed and parallel databases*, vol. 14, no. 1, pp. 5–51, 2003.

[45] P. Wohed, W. Van der Aalst, M. Dumas, A. Ter Hofstede, and N. Russell, "On the suitability of BPMN for business process modelling," *Business Process Management*, pp. 161–176, 2006.

[46] P. Wohed, W. Aalst, M. Dumas, and A. Hofstede, "Analysis of web services composition languages: The case of BPEL4WS," *Conceptual Modeling-ER 2003*, pp. 200–215, 2003.

[47] N. Russell and A. ter Hofstede, "Surmounting BPM challenges: the YAWL story," *Computer Science-Research and Development*, vol. 23, no. 2, pp. 67–79, 2009.

[48] E. Adamides and N. Karacapilidis, "A knowledge centred framework for collaborative business process modelling," *Business Process Management Journal*, vol. 12, no. 5, pp. 557–575, 2006.

[49] W. van der Aalst, A. ter Hofstede, and M. Weske, "Business process management: A survey," *Business Process Management*, pp. 1019–1019, 2003.

[50] D. Chappell, "Introducing Microsoft Windows Workflow Foundation: An early look," *MSDN Article, August*, 2005.

[51] OASIS, "Web services business process execution language (WS-BPEL) version 2.0." http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html, last accessed 25/03/2010.

[52] W. Van der Aalst, "Don't go with the flow: Web services composition standards exposed," *IEEE intelligent systems*, vol. 18, no. 1, pp. 72–76, 2003.

[53] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, "Examining the challenges of scientific workflows," *IEEE Computer*, vol. 40, no. 12, pp. 24–32, 2007.

[54] N. Russell and W. van der Aalst, "Evaluation of the BPEL4People and WS-HumanTask Extensions to WS-BPEL 2.0 using the Workflow Resource Patterns," *BPM Center Report BPM-07-10, BPMcenter. org*, 2007.

[55] "Windows Workflow Foundation scenarios guidance: Human workflow." http://msdn.microsoft.com/en-us/library/cc709416.aspx, last accessed 9/04/2010.

[56] R. Liu and A. Kumar, "An analysis and taxonomy of unstructured workflows," *Business Process Management*, pp. 268–284, 2005.

[57] P. Kammer, G. Bolcer, R. Taylor, A. Hitomi, and M. Bergman, "Techniques for supporting dynamic and adaptive workflow," *Computer Supported Cooperative Work (CSCW)*, vol. 9, no. 3, pp. 269–292, 2000.

[58] Y. Han, A. Sheth, and C. Bussler, "A taxonomy of adaptive workflow management," in *Workshop of the 1998 ACM Conference on Computer Supported Cooperative Work*, 1998.

[59] M. Koning, C. Sun, M. Sinnema, and P. Avgeriou, "VxBPEL: Supporting variability for Web services in BPEL," *Information and Software Technology*, vol. 51, no. 2, pp. 258–269, 2009.

[60] R. Filman, T. Elrad, and S. Clarke, *Aspect-oriented software development.* Addison-Wesley Professional, 2004.

[61] A. Charfi and M. Mezini, "Aspect-oriented workflow languages," *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, pp. 183–200, 2006.

[62] A. Erradi and P. Maheshwari, "AdaptiveBPEL: a policy-driven middleware for flexible web services composition," *Proceedings of Middleware for Web Services (MWS)*, 2005.

[63] "Workflow Patterns." http://www.workflowpatterns.com/, last accessed 25/03/2010.

[64] W. Van der Aalst *et al.*, "The application of petri nets to workflow management," *Journal of Circuits Systems and Computers*, vol. 8, pp. 21–66, 1998.

[65] W. Van der Aalst and A. Ter Hofstede, "YAWL: yet another workflow language," *Information Systems*, vol. 30, no. 4, pp. 245–275, 2005.

[66] A. ODE, "BPEL Simplified Syntax (simBPEL)." http://ode.apache.org/bpel-simplified-syntax-simbpel.html, last accessed 9/04/2010.

[67] M. Bischof, O. Kopp, T. Van Lessen, and F. Leymann, "BPELscript: A Simplified Script Syntax for WS-BPEL 2.0," in *35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2009)*, Citeseer, 2009.

[68] M. Blow, Y. Goland, M. Kloppmann, F. Leymann, G. Pfau, D. Roller, and M. Rowley, "BPELJ: BPEL for Java, A joint white paper by BEA and IBM," *BEA and IBM, March*, 2004.

[69] H. Smith, "Enough is enough in the field of BPM: We don't need BPELJ: BPML semantics are just fine," *unpublished paper, BPM3*, 2004.

[70] F. DeRemer and H. Kron, "Programming-in-the large versus programming-in-the-small," in *Proceedings of the international conference on Reliable software*, p. 121, ACM, 1975.

[71] A. Shalloway and S. Bain, "Programming by intention." http://www.netobjectives.com/resources/articles/programming-by-intention, last accessed 9/04/2010, 10 2008.

[72] D. Chappell, "Understanding BPM servers," *Microsoft White Paper*, 2004.

[73] "WebSphere Business Integration Server Foundation." http://www-01.ibm.com/software/integration/wbisf/, last accessed 9/04/2010.

[74] "Oracle® WebLogic Integration." http://download.oracle.com/docs/cd/E14981_01/wli/doc
last accessed 9/04/2010.

[75] "BizAgi BPM Suite." http://www.bizagi.com/, last accessed 9/04/2010.

[76] D. Chappell, "The workflow way: Understanding Windows Workflow Foundation." http://msdn.microsoft.com/en-us/library/dd851337.aspx, last accessed 9/04/2010, April 2009.

[77] "Google Wave." http://wave.google.com/, last accessed 9/04/2010.

[78] G. Trapani and A. Pash, "The complete guide to Google Wave," 2010.

[79] "Update on Google Wave." http://googleblog.blogspot.com/2010/08/update-on-google-wave.html.

[80] A. Dreiling, "Gravity – collaborative business process modelling within Google Wave." http://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/15618, last accessed 9/04/2010, September 2009.

[81] Itensil, "Workflow-on-Wave." http://itensil.com/demos/, last accessed 9/04/2010.

[82] "Wikipedia." http://www.wikipedia.org/, last accessed 9/04/2010.

[83] M. Czuchra, "Oryx: Embedding business process data into the web," *Final bachelor's paper Hasso Plattner Institute at the University of Potsdam*, 2007.

[84] M. Held and W. Blochinger, "Collaborative bpel design with a rich internet application," in *8th IEEE International Symposium on Cluster Computing and the Grid*, pp. 202–209, 2008.

[85] Lombardi, "Blueprint." http://blueprint.lombardi.com/, last accessed 9/04/2010.

[86] "mxGraph – JavaScript diagram component by JGraph." http://www.jgraph.com/mxgraph.html, last accessed 9/04/2010.

[87] J. D. Eisenberg, *SVG Essentials*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2002.

[88] J. Schiller, "SVG browser support table." http://www.codedread.com/svg-support.php, last accessed 9/04/2010.

[89] MSNBC, "Creator of web spots a flaw in Internet Explorer." http://www.msnbc.msn.com/id/26646919/, last accessed 9/04/2010.

[90] S. White, "Introduction to BPMN," *IBM Cooperation*, pp. 2008–029, 2004.

[91] "Facebook." http://www.facebook.com, last accessed 8/9/2010.

[92] "World Wide Web Consortium (W3C)." http://www.w3.org/, last accessed 8/9/2010.

[93] D. Crockford, "JSON: The fat-free alternative to XML," in *Proc. of XML*, vol. 2006, 2006.

[94] "Batik SVG Toolkit." http://xmlgraphics.apache.org/batik/, last accessed 10/9/2010.

[95] M. Owens and M. Owens, *The definitive guide to SQLite.* Apress, 2006.

[96] "Prototype JavaScript framework." http://www.prototypejs.org/, last accessed 8/9/2010.

[97] "script.aculo.us, Web2.0 JavaScript library." http://script.aculo.us/, last accessed 8/9/2010.

[98] "jQuery Plugins: SVG Integration." http://plugins.jquery.com/project/svg, last accessed 10/9/2010.

[99] "Raphaël JavaScript library." http://raphaeljs.com/, last accessed 10/9/2010.

[100] "Cascading style sheets." http://www.w3.org/Style/CSS/, last accessed 8/9/2010.

[101] M. La Rosa and S. Clemens, "The order fulfillment process (in YAWL)," 2009.

[102] "The Knowledge and Data Engineering Group (KDEG)." http://kdeg.cs.tcd.ie/, last accessed 6/9/2010.

[103] J. Brooke, "SUS-A quick and dirty usability scale," *Usability evaluation in industry*, pp. 189–194, 1996.

[104] B. Walsh, "Tools and groups for startups," *The Web Startup Success Guide*, pp. 99–143, 2009.

[105] "OWASP (Open Web Application Security Project) Top 10 2007." http://www.owasp.org/index.php/Top_10_2007, last accessed 9/04/2010.

[106] M. Howard and D. Leblanc, *Writing secure code.* Microsoft Press Redmond, WA, USA, 2002.