Information Systems 000 (2017) 1–17

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Near real-time collaborative modeling for view-based Web information systems engineering

Petru Nicolaescu[a,*], Mario Rosenstengel[a], Michael Derntl[c], Ralf Klamma[a], Matthias Jarke[a,b]

[a] RWTH Aachen University, Lehrstuhl Informatik 5, Ahornstr. 55, Aachen, Germany
[b] Fraunhofer FIT, Birlinghoven Castle, Sankt Augustin, Germany
[c] Eberhard Karls Universität Tübingen, eScience-Center, Wilhelmstr. 32, Tübingen, Germany

ARTICLE INFO

Article history:
Received 1 November 2016
Revised 31 March 2017
Accepted 31 July 2017
Available online xxx

MSC:
00-01
99-00

Keywords:
Metamodeling
Views
Viewpoints
Collaborative modeling

ABSTRACT

Conceptual modeling is a creative, social process driven by the views of stakeholders. In modern, agile development – especially for continuously evolving Web applications – contributions from a wide variety of geographically distributed stakeholders, their involvement in negotiation and impact analysis from different perspectives and the rapid prototype generation from specifications gain much importance. Moreover, people have come to expect easy near real-time system support with few restrictions. While conceptual modeling or CSCW environments exist for each of these individual aspects, their interplay has barely been studied. This paper presents a collaborative conceptual modeling approach called SyncMeta that aims to close this gap by supporting view-based modeling in the Web browser in a near real-time shared editing environment[1]. In addition to domain-specific graphical modeling languages, viewpoints can be collaboratively defined on the meta-modeling layer and instantiated as views within a model editor instance. Besides this formal structure, we also discuss the choice of algorithms for locking-free shared model editing and of highly scalable team size. In addition to this kind of technical evaluation, the paper also presents several medium-scale user studies that have accompanied the iterative SyncMeta development. These studies investigated both the requirements (quality advantages and user acceptance of the view-based approach, near real-time support) and important design options such as centralized vs. peer-to-peer viewpoint resolution. A fully functional SyncMeta software framework which incorporates most of these results, is available in the GitHub open source repository. We expect this line of work to pave the way for methodologies and tools for socially engineered Web information systems.

1. Introduction

Conceptual modeling plays an important role in representing domain-specific information during the requirements elicitation and design phases of information systems [2]. Conceptual modeling frameworks reduce system complexity by introducing abstraction levels and by offering sublanguages for different static, dynamic, or goal-oriented perspectives, which allow involved stakeholders to focus on artifacts and concepts of their interest while linking these perspective models through formal metamodels. In addition, the concept of **views** allows stakeholders to focus their attention on modeling areas of interest within or across modeling perspectives. Previous research around the turn of the century formally investigated stakeholder viewpoints and the generation of views from metamodels particularly from a requirements engineering perspective [2,3]. For example, certain stakeholders may prefer a different view to express their concerns [4,5], as models in complex systems tend to become quite large.

Views are used to deal with this complexity in (meta)modeling frameworks. View-based model-driven design aims at creating partial metamodels and models, each reflecting a set of concerns of the modeled system [4]. Of course, collaborative development using views raises the need for conflict resolution between possibly conflicting viewpoints within or across modeling perspectives. In addition, there is the challenge of generating coherent code satisfying the requirements specified in an agreed set of views, in order to enable model-driven generation in later development stages, as well as in system evolution.

However, the world of software development has been moving on in the last decade. Compared to model-driven approaches in classical software development methodologies, conceptual

* Corresponding author.
E-mail address: nicolaescu@dbis.rwth-aachen.de (P. Nicolaescu).

[1] This paper is an extension of Nicolaescu et al. [1], "View-based Near Real-Time Collaborative Modeling for Information Systems Engineering", in CAiSE 2016 (best paper award).

http://dx.doi.org/10.1016/j.is.2017.07.008
0306-4379/

modeling plays a less prominent role in agile methodologies where end users are actively and continuously involved in application design and development (i.e., end user development [6], participatory design [7]). This is of particular importance in Web information systems engineering that is often characterized by buzzwords such as “perpetual beta” or “innovation from the edge”. Together with suitable platforms, agile Web information systems development represents a suitable driver for rapid uptake of technological advances and the re-design of software landscapes into agile modern systems [8].

In such a Web-based setting, from a social perspective, conceptual modeling and agile processes are often the result of collaboration between team members, which may not be geographically co-located, yet are accustomed from their usual Web tools that everything has to happen in **near real-time** (NRT), by which in this paper we mean to say “almost always without human-noticeable delay though less stringent than real-time requirements in embedded systems”. In contrast, most previous computer-aided software engineering and modeling tools have supported asynchronous collaboration, using mechanisms such as locking, serialized file access, versioning techniques, email exchange, etc.) [9]. Through this rather strict governance, they rarely considered the desire and potential advantages for unobstructed, synchronous shared editing [10,11].

Fortunately, recent advanced Web protocols such as WebRTC [12], WebSockets [13], XMPP [14] have begun to offer scalable, cheap and ubiquitous technical means for unobstructed, near real-time collaboration and synchronous communication. This raises the opportunity to re-investigate the collaborative conceptual modeling field from the NRT perspective, in order to address challenges such as (formally controlled) agile development for large-scale Web information systems. We aim to offer better support for software development processes of multi-disciplinary applications ranging from social software to Industrial Internet in the production and logistics domain. Among other goals, this re-opens challenges from the modeling literature such as metamodeling, metamodel to model generation, or nudging and recommendations during the model authoring process.

In this paper, we specifically explore the opportunities and challenges of integrating view abstraction and view-based model-driven development with the NRT setting. The proposed SyncMeta approach allows the collaborative creation of metamodels in NRT based on a visual language specification and the generation of collaborative model editors from the defined metamodels. Moreover, it facilitates a collaborative, graphical definition of views on the metamodel layer which can then be applied in models. We have studied alternative algorithms for NRT conflict resolution, as well as selection criteria and specific alternatives for suitable user interface toolkits. In several stages of evaluation during an iterative development process of our approach, technical criteria such as flexibility, scalability, and NRT performance have been considered. This has been complemented by user studies in which we asked questions such as “should the view-based approach be constructed with a more centralistic model-view linkage, or with direct peer-to-peer (view-to-view) mappings?”, or “how important is the NRT aspect to development teams?”. Altogether five such case studies with different kinds of modelers were conducted in parallel to the iterative development process of SyncMeta [15] a Web-based framework which provides a flexible and performant architecture for achieving domain independent collaborative modeling, and is available in the GitHub repository[2].

The remainder of this paper is structured as follows. Section 2 introduces a motivating scenario from Web information system engineering, from which the main requirements of our NRT collaborative conceptual modeling approach are derived. Section 3 presents the collaborative modeling process involving a metamodel editor, a model editor and collaborative view-based modeling features. Section 4 provides a formalization of view-based metamodeling. Section 5 and 6 detail the abstraction hierarchy and describe how the metamodeling layer of SyncMeta helps to generate viewpoints, resp. their implementation in the SyncMeta model editor. Section 7 presents the NRT collaboration and shared editing perspective of our visual modeling approach. Section 8 then describes the architecture and implementation of the SyncMeta framework and discusses the limitations. Section 9 presents the user evaluation strategy and overall results; moreover, it elaborates one case study in detail. Section 10 compares our approach to the state-of-the-art. Finally, Section 11 summarizes our conclusions and outlines future work.

2. Scenario and requirements

The research reported in this paper is inspired from a model-driven Web engineering scenario where developers and end users collaboratively build Web applications [16]. Fig. 1 showcases such a scenario. Given a set of elicited requirements in a joint project, multiple stakeholders (e.g., end users, software engineers, architects and developers) want to use agile methods to design and develop a Web information system. A first prototype should be released as soon as possible. The various stakeholders establish a remote meeting session, where they can agree on a specification. During this session, they use a collaborative modeling platform to design the frontend and discuss and implement the needed functionality. Using this platform, models can be authored collaboratively in NRT using dedicated views between multiple users. As such, end users, designers and developers can co-design the frontend of the application using the frontend view. All modelers are aware of the remote actions of their collaborators and are connected using chat, video and voice communication channels. Assisted by developers, end users and designers add on the modeling canvas the frontend components which they envision in the final prototype. Developers and other stakeholders with technical knowledge model in NRT the backend of the information system (services, database, service interface, etc.) in a separate view on the model offered by the editor. Finally, to compose and release the final application, software architects are interested in a view which can depict the software components and the links between them. The Web information system can be constructed in this mashup view, where stakeholders can model the communication between components. The system is generated and deployed using the resulting model containing all elements authored in the three views.

In order to use such a visual modeling language editor, it first needs to be defined. The domain-specific modeling languages used during the collaborative sessions in the editor must therefore be previously specified. To this purpose, involved stakeholders (i.e., developers, architects, software engineers) - using the same collaborative modeling platform - define a metamodel that specifies visually the custom modeling language for building a Web information system. During the metamodeling process, they also create using metamodel entities certain viewpoints of the information system: one for frontend elements, one for the backend and a third for modeling the communication/mashup and orchestration of the developed frontend and backend elements. Once the metamodel and viewpoints are defined, the model instance editor can be generated and used as shown in Fig. 1.

The focus of this paper is to create a lightweight collaborative view-based conceptual modeling approach that can deal with multiple modelers working in NRT. The main **functional requirements** for obtaining a view-based framework oriented towards collaboration on the Web (identified while examining scenarios like the one

[2] https://github.com/rwth-acis/syncmeta.

Please cite this article as: P. Nicolaescu et al., Near real-time collaborative modeling for view-based web information systems engineering, Information Systems (2017), http://dx.doi.org/10.1016/j.is.2017.07.008