Available online at www.sciencedirect.com
ScienceDirect
Information and Software Technology 50 (2008) 296–321
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

On the interplay between inconsistency and incompleteness in multi-perspective requirements specifications

Ana Belén Barragáns Martínez *, José J. Pazos Arias, Ana Fernández Vilas, Jorge García Duque, Martín López Nores, Rebeca P. Díaz Redondo, Yolanda Blanco Fernández

Telematic Engineering Department, University of Vigo, Vigo, Spain

Received 23 March 2006; received in revised form 22 January 2007; accepted 4 February 2007
Available online 13 February 2007

Abstract

A major challenge for dealing with multi-perspective specifications, and more concretely, with merging of several descriptions or views is toleration of incompleteness and inconsistency: views may be inconclusive, and may have conflicts over the concepts being modeled. The desire of being able to tolerate both phenomena introduces the need to evaluate and quantify the significance of a detected inconsistency as well as to *measure* the degree of conflict and uncertainty of the merged view as the specification process evolves.

We show in this paper to what extent disagreement and incompleteness are closely interrelated and play a central role to obtain a measure of the level of inconsistency and to define a merging operator whose aim is getting the model which best reflects the combined knowledge of all stakeholders. We will also propose two kinds of interesting and useful orderings among perspectives which are based on differences of behavior and inconsistency, respectively.

Keywords: Viewpoints; Uncertainty; Inconsistency; Requirements specification; Merging

1. Introduction

There are many people involved with the development of a large software system, both in its design and use. Whilst these people can be broadly classed into groups, such as users (often of several types), management and software developers, these groupings will not enforce conformity of the individual members. Each person is a potential source of knowledge. In requirements elicitation, inconsistencies occur frequently, usually indicating a conflict between the interested parties. In such cases, the conflict represents the need for an explicit decision by the analyst, which should not be taken until all the appropriate information has been gathered, since as far as possible, all alternatives must be captured and accommodated, the project timetable must allow the analyst to delay these decisions as far as is necessary.

An effective way to specify individual concerns is via viewpoints-based approaches [1–4]. Reasoning from different *viewpoints* is a necessary part of most design processes. This task often involves collecting information from a number of potentially conflicting perspectives or sources, or participants with different views, and forming a single combined view or perspective (a synthesis, or consensus [5]). Consensus building is essential in aligning multiple stakeholder viewpoints which are commonly emerging during the process of requirements elicitation, analysis and validation. In fact, agreement among diverse groups of stakeholders is deemed prerequisite to establishing project cooperation and collaboration.

* Corresponding author. Address: Departamento de Enxeñería Telemática, E.T.S.E. Telecomunicación, Vigo 36310, Spain. Tel.: +34 986 813473; fax: +34 986 812116.
E-mail address: belen@det.uvigo.es (A.B. Barragáns Martínez).
URL: http://www.det.uvigo.es/~belen (A.B. Barragáns Martínez).

0950-5849/$ - see front matter
doi:10.1016/j.infsof.2007.02.001

Moreover, research into group behavior indicates that conflict can produce higher quality solutions. Certainly, exploration of the areas where participants descriptions differ can lead to a much better understanding of the domain. This is a strong argument for conflict to be carefully managed in the software process, with participants encouraged to express divergent views. This will ensure that the resulting system does not reflect just one point of view, and does not ignore concerns which interfere with the dominant concern. So the ultimate goal of the requirement process should be to produce a specification which represents all concerns.

On the other hand, as several parallel specifications evolve, they will frequently conflict with each other, and at all times they will be (to some degree) incomplete. At times there will be temporary inconsistencies, overgeneralizations, and over-simplifications. However, participants will need to manipulate each specification as it evolves, as part of the exploratory process, and so the reasoning mechanisms must cope with disagreement and incompleteness. Areas of conflict, and places where more details are needed can often be detected automatically, but the need to allow commitments to be delayed means that participants might choose not to resolve these immediately.

With the motivation of being able to obtain a specification which represents all concerns as it evolves and, therefore, being able to cope with disagreement and incompleteness, our main aim consists of getting, at intermediate stages of the development process, a merged view which properly reflects the knowledge of each participant in the elicitation tasks. For this purpose, we have identified several challenges to be addressed by our methodology, referred to as MultiSpec (Multi-Perspective Methodology for Software Requirements Specifications, overviewed in next section), which has been devised to support the evolution of software requirements specifications gathered from multiple perspectives, formalizing the reasoning in presence of conflicts (overspecification) and incompleteness (underspecification) making use of an underlying many-valued logic.

One of these challenges, as expected, is the ability to evaluate and quantify the significance of a detected inconsistency. Throughout this paper, we use the term *inconsistency* to refer to a situation in which there is no total agreement between all involved parties. The reason of the appearance of this inconsistency can be twofold: it might be due to some kind of disagreement (some stakeholders have contradictory opinions) and/or it might be due to the presence of some uncertainty (some stakeholders have not decided yet because, for example, they have not enough information to do it). Moreover, the terms *conflict* and *disagreement* as well as *uncertainty* and *incompleteness* can, respectively, be exchanged throughout the article. The same happens with the terms *view*, *viewpoint* and *perspective* as well as with *stakeholder* and *agent*, unless otherwise stated.

We show in this paper to what extent incompleteness and disagreement are closely interrelated and play a central role to obtain a measure of the significance of a detected inconsistency and to define a merging operator whose aim is getting the model which best reflects the combined knowledge of all stakeholders. Moreover, after having defined a preliminary inconsistency metric in [6] and having investigated on composition operators in [7,8], we have realized that we also need to measure the *total inconsistency of the merged view* as the specification evolves, to check, for example, how the degree of inconsistency evolves in each cycle. Once again, the influence of uncertainty and conflict on the measure of total inconsistency of the merged model will be made patent.

Our last contribution consists of introducing two partial orderings. The first one is useful to compare different merged models and is based on the measure of the total inconsistency. This inconsistency-based ordering allows us to measure the degree of evolution of the merged view at each step of the development process. The second one is a closeness relation based on differences of behavior which permits us to compare different perspectives and see which of them is closer to the majority thinking, and therefore, less inconsistent with the stakeholders' global opinion.

The paper is organized as follows. After this introduction, Section 2 gives an overview of the MultiSpec methodology which is needed to understand the context of this work, as well as motivating the paper and introducing some preliminary definitions. Section 3 is a piece of the main part of this paper where we define and explain in detail the metric proposed to evaluate the level of significance of each detected inconsistency. Section 4 completes the core of this work where we show, respectively, the merging algorithm and the proposed merging operator. In Section 5, the two orderings, one based on inconsistency and the other based on behavior, are presented and explained. Section 6 illustrates the issues which were dealt with in previous sections through a little example. Section 7 surveys and discusses the related work in areas addressed by this paper: techniques for handling inconsistency and approaches for merging views. Finally, Section 8 concludes and discusses directions for future work, some of which we are currently undertaking.

2. Overview of MultiSpec and motivation

MultiSpec makes use of a dual scheme to represent each viewpoint: model- and property-oriented specification. First, as property-oriented, a many-valued temporal logic SCTL (Simple Causal Temporal Logic) is proposed to elicit the requirements of each stakeholder. Second, as model-oriented, the methodology proposes a state-transition formalism MUS (Model of Underspecified States) to internally represent each viewpoint because the chosen representation scheme must also be amenable to various forms of reasoning and analysis. Therefore each