Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER Electronic Notes in Theoretical Computer Science 133 (2005) 275–294
www.elsevier.com/locate/entcs

Model Checking Publish/Subscribe Notification for thinkteam®

Maurice H. ter Beek[a] Mieke Massink[a] Diego Latella[a]
Stefania Gnesi[a] Alessandro Forghieri[b] Maurizio Sebastianis[b]

[a] *Istituto di Scienza e Tecnologie dell'Informazione, CNR*
Area della Ricerca di Pisa, Via G. Moruzzi 1, 56124 Pisa, Italy
{maurice.terbeek,mieke.massink,diego.latella,stefania.gnesi}@isti.cnr.it

[b] think3, *Inc.—European Headquarters, Via Ronzani 7/29, 40033 Bologna, Italy*
{alessandro.forghieri,maurizio.sebastianis}@think3.com

Abstract

This paper reports on the fruitful combination of academic experience with formal modelling techniques and industrial experience with requirements exploration. We study the addition of a publish/subscribe notification service to thinkteam,[a] a ready-to-use Product Data Management application developed by think3. thinkteam allows enterprises to capture, organise, automate, and share engineering product information and it is an example of an asynchronous and dispersed groupware system. We define an abstract specification (model) of the groupware protocol underlying thinkteam and augment it with a publish/subscribe notification service. Consequently, we show a number of important correctness properties of the thinkteam model, some of which are also relevant to groupware protocols in general. In particular, we show that by adding a publish/subscribe notification service to thinkteam, the user's awareness of the status of the development of the engineering product and the activities of the design team increases.

[a] thinkteam is a registered trademark of think3, Inc. For details: http://www.think3.com.

Keywords: publish/subscribe notification, thinkteam, model checking, groupware, awareness, concurrency control.

1 Introduction

Computer Supported Cooperative Work (CSCW) is an interdisciplinary research field which deals with the understanding of how people work together, and the ways in which computer technology can assist them [13]. This technology mostly consists of multi-user computer systems called groupware (sys-

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2004.08.069

tems) [1,10]. Groupware is typically classified according to two dichotomies, viz. (1) whether its users work together at the same time (synchronous) or at different times (asynchronous) and (2) whether they work together in the same place (co-located) or in different places (dispersed). This is called the time space taxonomy by Ellis et al. [10]. In this paper we deal with thinkteam, which is an asynchronous and dispersed groupware system. Other examples include electronic mail, workflow, collaborative writing systems, and the version-control systems often used in software engineering to coordinate the changes made by multiple programmers to the same program.

Some important design issues in groupware systems are data sharing, user awareness, and concurrency control. In this paper we address these issues in the context of thinkteam. More precisely, we use model checking to formalise and verify a number of properties specifically of interest for the correctness of groupware protocols in general, i.e. not limited to the context of thinkteam. In recent years there has been an increasing interest in the use of model checking for the formal verification of (properties of) groupware [3,17,21] and publish/subscribe (pub/sub) systems [6,7,12,20].

thinkteam is think3's Product Data Management (PDM) application catering the product/document management needs of design processes in the manufacturing industry. Its main strengths are a rapid deployment and startup cycle, its flexibility, and a seamless integration with thinkdesign—think3's CAD solution—as well as with other third party products. thinkteam allows enterprises to manage the capturing, organising, automating, and sharing of engineering product information in an efficient way. In this paper we study the addition of a lightweight and easy-to-use pub/sub notification service to thinkteam. The goal of adding such a service to an application is to increase user awareness by intelligent data sharing: whenever a user publishes a document by sending it to a centralized repository, automatically all users that are subscribed to that document are asynchronously notified via a multicast communication. Due to a potentially large number of users, it is fundamental to use subscription-based multicast communication rather than broadcast communication. Other examples of applications of a pub/sub notification service include electronic auctions on the Internet and email alert services for new journal or book releases that many publishing houses offer nowadays.

Pub/sub notification decouples the communication among users: a user that publishes a document need not be concerned with whom the server will send a notification to, i.e. the users communicate through the server. Users need not actively participate in the notification in a synchronous way. In fact, the main strength of a pub/sub notification service is said to be the "full decoupling of the communicating participants in time, space and flow" [11].