The thinkteam® Case Study: Overview and Outlook

Maurice H. ter Beek

FM&&T, ISTI-CNR

thursday 9 December 2004

Workshop SP4

ISTI-CNR

⇒ this is joint work with Mieke Massink, Diego Latella, and Stefania Gnesi from FM&&T, and Alessandro Forghieri and Maurizio Sebastianis from think3

Outline

- recalling the **thinkteam** case study
- adding publish/subscribe notification
- verifying by means of model checking
- publications, conclusions & future work

Recalling thinkteam (TT)

Product Data Management (PDM) application

A dispersed & asynchronous groupware system

Provides PDM needs of design processes in the manufacturing industry

Strengths: rapid deployment & startup cycle, flexible, smooth integration with **thinkdesign** (think3's CAD solution) & 3rd party products

Helps to capture, organise, automate & share engineering product information efficiently

Controlled storage and retrieval of document data in PDM applications is called *vaulting*

Adding Publish/Subscribe Notification

Raise user awareness by intelligent data sharing:

"whenever a user *publishes* a document by sending it to the vault, automatically all users that are *subscribed* to that document are notified via an asynchronous multicast communication"

Notion recently much studied in the literature !

+ "full decoupling of the communicating participants in time, space & flow" [EFGK03]

− generally difficult to verify [GKK03,ZGB03]

⇒ we have formally modelled & verified the addition of publish/subscribe notification to TT [BMLGFS04a,BMLGFS04b,BMLGFS04c]

Correctness Criteria of TT Protocol

Concurrency Control

1) every lock request is eventually answered
2) only one user at a time may possess a lock on a file
3) every file lock is eventually released
4) a file lock is not released after a *checkInOut*

Awareness

1) no user receives (a) a *notify* or (b) an *update* if not registered
2) every *checkOut* or *checkInOut* eventually leads to a *notify* to all (and only those) registered users
3) every *unCheckOut*, *checkIn*, or *checkInOut* eventually leads to an *update* to all (and only those) registered users

Denial of Service

1) no user is forever denied a service

Main results of verification

All verifications were performed by running Spin Version 4.1.3 on a SUN Netra X1 workstation with $1,000$ Mb of available physical memory

+ full statespace search $\Rightarrow$ no deadlock states

+ concurrency control & awareness aspects of the TT protocol augmented with publish/ subscribe notification largely well designed

− 'superuser' required to force a user to ever return a checked out document to the vault

− document reservation system required (rather than access based on the 'retrial' principle) to avoid a user endlessly keeps busy the CC

Publications

ter Beek-Massink-Latella-Gnesi-Forghieri-Sebastianis :

"A Case Study on the Automated Verification of Groupware Protocols"

(accepted for the Experience Reports Track of ICSE'05—the 27th International Conference on Software Engineering, ACM)

"Model Checking Publish/Subscribe Notification for **thinkteam**"

(FMICS'04—9th International Workshop on Formal Methods for Industrial Critical Systems, ENTCS, Elsevier)

"Automated Verification of Groupware Protocols"

(ERCIM News Special: Automated Software Engineering 58, 2004)

used experience from ter Beek-Massink-Latella-Gnesi :

"Model Checking Groupware Protocols"

(COOP'04—6th International Conference on the Design of Cooperative Systems, IOS Press)

Conclusions

- case study on formalisation & verification of concurrency control & distributed notification aspects of groupware protocol underlying TT

- show feasibility & usefulness of model checking when verifying groupware protocols in general

- among first successful applications of exhaustive model checking to verification of publish/subscribe notification in a groupware setting

- think3 intends to use specification as basis for planned implementation of such services in TT

- think3 expressed interest in acquiring the skills to apply automated verification to the (groupware) protocols that underlie their software

Future Work

- uptake by other research groups: Lubos Brim's Parallel & Distributed Systems Lab (Masaryk U., Brno, Czech Rep.)–distributed model-checking

- integrated formal approach to system analysis covering functional & non-functional aspects

- use (a)CSL for formal characterisation & verification of functional & non-functional properties

- explore feasibility of *qualitative* & *quantitative* analysis through stochastic model checking

$\Rightarrow$ no straightforward application of technology:

  - mostly using tools still in prototype phase
  - application in this industrial setting is new
  - technology transfer aspect is challenging

- apply acquired knowledge & experience to **thinkteam** & other (groupware) protocols