Formal Models for Distributed Negotiations

Introduction

Thanks

- Local organization and PC, especially
  - Prof. Enrique Carlos Segura
  - Mrs. Aida Gabriela Interlandi
- Help with preparation of electronic material
  - Dr. Hernan Melgratti
- Research support
  - FET-GC Project AGILE
  - Italian MIUR Project CoMeta
Audience

- First time the course is proposed
- Feedback is always important
- Most slides are newly prepared
  - Corrections and suggestions are welcome
  - Please do not hesitate to ask questions
  - Too slow, too fast, too much?
- Language
  - English / Italian / Castellano?

Revised Abstract I

- In the area of concurrent, distributed and mobile languages there is a renewed interest toward models, languages and primitives for orchestration
- In particular, for largely distributed scenarios like WAN computing, web programming, and, more generally, global computing, where synchronous communication is unrealistic, the needs of primitives for handling contract stipulation, distributed agreements, causally dependent decisions, negotiations with nested choice points to be carried out concurrently emerges as a key issue in most commercial applications.
Revised Abstract II

- Such applications can run on different platforms and require coordination layers between components that are designed and implemented separately (e.g. e-commerce or on-line auction systems).
- Though orchestration can be interpreted as the atomic execution of certain control activities (e.g. in terms of transactions), more generally the problem is that of committing the results of long-running distributed decision processes as soon as the participants reach partial agreements.

Revised Abstract III

- Current research aims at finding some fully distributed models and languages that can provide convenient alternatives to the centralized (transaction) managers that are usually employed in running applications.
- Still, one important aspect that should be guaranteed is the possibility of viewing modelled systems at different level of abstraction.
Scenario

- Largely distributed systems
  - Wide Area Network (WAN)
  - Global Computing (GC)
  - Web programming and web services
- Asynchronous communication
- Transactions, contracts, negotiations, decisions, agreements, choices
  - Causality, concurrency and distribution
- Language primitives and formal models
  - Coordination, orchestration, choreography

Examples

- Vacation booking
  - Flights, hotels, cars, theatres
- Business process description
  - Deliveries, payments, penalties, subcontracts
- Data processing applications
  - File systems, updating, modifications
- Reverse auction systems
  - Dynamic pricing, one buyer many sellers
- Banking transactions
- Mobile lessees
Outline

- Different flavours and extensions of:
  - Transactions
  - Commit protocols
  - Workflows
  - Petri Nets (PN)
  - Process Description Languages
- Examples, comparisons and ongoing work
- Expressiveness and feasibility

Part I

- The 1st part (3 hours) will illustrate the main aspects to be taken into account by formal models, e.g.
  - atomicity
  - isolation
  - nested commits and aborts
  - compensation
  - dynamicity
  - distribution
  - ...
Part II

The 2nd part (10 hours) will survey some recent proposals in the literature, based on suitable extensions of
- Petri nets
- Linda
- CCS
- join-calculus
- pi-calculus
- commit protocols
- compensation algorithms
- ...

Part III

The 3rd and last part (2 hours) will sketch some ongoing work in combining all the ingredients to define a flexible language for distributed negotiations
- Committed join-calculus
Origins

- Collaborations
  - Ugo Montanari, Hernan Melgratti (Pisa)
  - Cosimo Laneve (Bologna)
- Related work
  - Nadia Busi, Gianluigi Zavattaro, Roberto Gorrieri, Laura Bocchi (Bologna)
  - Marzia Buscemi (Pisa)
  - Vladimiro Sassone (Brighton)
  - Martin Berger, Kohei Honda (London)
  - Tony Hoare, Carla Ferreira, Dominic Duggan, ...

Material

- Lectures slides
  - MS Power Point presentations
  - Separated by arguments
- Selected papers
  - Partially distributed as course notes
  - Mostly available on-line
  - Referenced by arguments in presentations
Prerequisites

- Basic knowledge of mathematic and logic
- Helpful but not mandatory:
  - Petri nets
  - Process description languages
    - CCS, Linda, join calculus, pi-calculus
    - Structural Operational Semantics
  - BizTalk, XLANG, StAC, BPEL, JavaSpaces, ...
- How many of you are familiar with ... ?

Outcome

- Some familiarity with a flourishing area of theoretical research and applications
  - Analysis of primitives and models
  - Broad panoramic of existing proposals
  - Experience with key aspects, techniques and ingredients
  - Mathematical basis for discerning features and expressiveness of different approaches
Exams

- Recommended exercises to gain familiarity with the arguments presented in the course
  - Assigned during the various lectures
    - To prove some properties
    - To encode some examples
  - Instructions for returning the solutions
    - Deadline: Monday 15/9/2003
    - At least 6 problems (of which at least one per day) solved
    - Individually solved (rating 1-10)
    - Must be prepared in portable electronic form (either postscript or PDF, please keep size small!)
    - Must be sent to bruni@di.unipi.it (ask for acknowledge)