Example of a Problem Statement: Introduction into ARENA

Outline

- Problem Statement
- Functional Requirements
- Nonfunctional Requirements
- User Interface
- Object Model
- System Decomposition
- Deployment

Problem Statement

- The problem statement is developed by the client as a description of the problem addressed by the system

- A problem statement describes
  - The current situation
  - The objectives
  - The functionality the new system should support
  - The environment in which the system will be deployed
  - Deliverables expected by the client
  - Delivery dates
  - A set of acceptance criteria.

Ingredients of a Problem Statement

- Current situation
  - The problem to be solved
  - Description of one or more scenarios
- Objectives
- Requirements
  - Functional and nonfunctional requirements
  - Constraints (“pseudo requirements”)
- Target environment
  - The environment in which the delivered system has to perform a specified set of system tests
- Project schedule
  - Major milestones including deadline for delivery
- Client acceptance criteria
  - Criteria for the system tests.

Current situation: The problem to be solved

- There is a problem in the current situation
  - Examples:
    - The response time when playing chess is too slow.
    - I want to play Go, but cannot find players on my level.
- What has changed? Why can address the problem now?
  - ***Change in the application domain***
    - A new function (business process) is introduced into the business
  - ***Change in the solution domain***
    - A new solution (technology enabler) has appeared

ARENA: The Current Situation

- The Internet has enabled virtual communities
- Multi-player computer games now include support for virtual communities
  - Players can receive news about game upgrades, new game levels, announcement of matches and scores
- Currently each game company develops such community support in each individual game
  - Each company uses a different infrastructure, different concepts, and provides different levels of support
- This redundancy leads to problems:
  - High learning curve for players joining a community
  - Game companies develop the support from scratch
  - Advertisers contact each community separately.

ARENA: The Objectives

- Provide a generic infrastructure to
  - Support virtual game communities.
  - Register new games
  - Register new players
  - Organize tournaments
  - Keeping track of the players scores.
- Provide a framework for tournament organizers
  - to customize the number and sequence of matchers and the accumulation of expert rating points.
- Provide a framework for game developers
  - for developing new games, or for adapting existing games into the ARENA framework.
- Provide an infrastructure for advertisers.

ARENA: The Objectives (2)

- Provide a framework for tournament organizers
  - to customize the number and sequence of matchers and the accumulation of expert rating points
- Provide a framework for game developers
  - for developing new games, or for adapting existing games into the ARENA framework
- Provide an infrastructure for advertisers.

ARENA Functional Requirements

- Spectators must be able to watch matches in progress without prior registration and without prior knowledge of the match
- The operator must be able to add new games.

ARENA Nonfunctional Requirements

- The system must support
  - 10 parallel tournaments,
  - Each involving up to 64 players
  - and several hundreds of spectators.
  - The ARENA server must be available 24 hours a day
- The operator must be able to add new games without modifications to the existing system
- ARENA must be able to dynamically interface to existing games provided by other game developers.

Constraints

- Constraint: Any client restriction on the solution domain and project management
  - Sometimes also called Pseudo Requirements
  - Constraints restrict the solution space
- Example of constraints
  - Delivery constraints ("must be delivered before Christmas")
  - Organizational constraints ("must have a separate testing team")
  - Implementation constraints ("must be written in Cobol")
  - Target platform constraints ("must run on Windows 98")

ARENA Target Environment

Example:

- Users must be able to run ARENA games as applets in any Web Browser
- The web page must be validated through the *W3C Markup Validation Service*
- Interaction with the ARENA Server must be via HTTP/1.1.

To be distinguished from development environment

- "Prototypes will be built with Revolution 2.6.1"
- "Games will be tested with Internet Explorer and Firefox"
- "The implementation language will be Java 1.4.2."
- "The IDE will be Eclipse 3.2"

Project Schedule

- The project schedule is an optional part of the problem statement
  - Managerial information
  - Often the seed for the schedule in the software project management plan.
- Lists only major milestones negotiated with the client
  - 3 to 4 dates (fixed dates!)
- Example:
  - Project-kickoff April 15
  - System review May 15
  - Review of first prototype Jun 10
  - Client acceptance test July 30

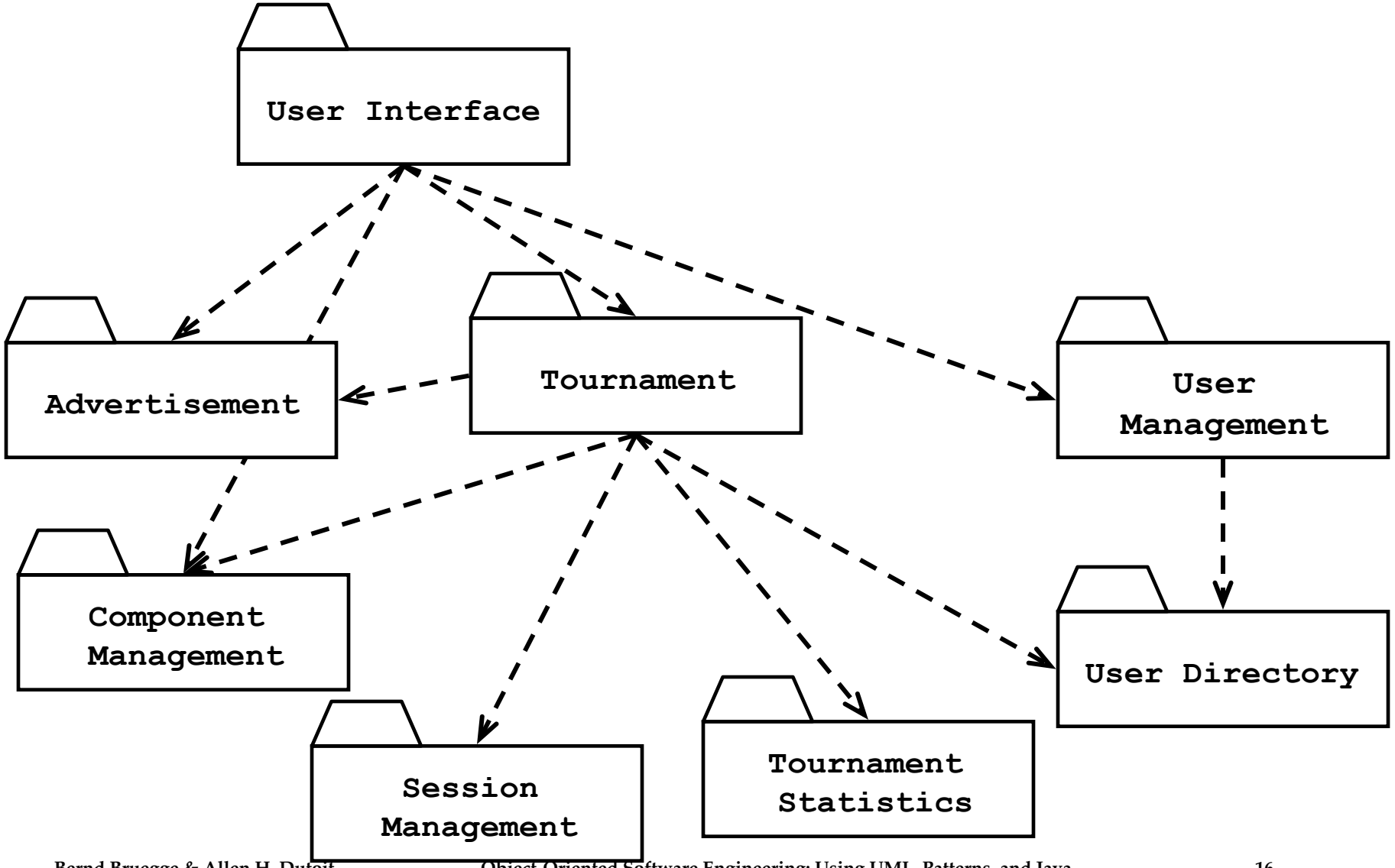
Client Acceptance Criteria

- The system supports 10 parallel tournaments with 64 players and 10 spectators per tournament
- The client supports the games Tic-Tac-Toe and Asteroids
- The average response time for a command issued by a client is less than 1 second
- The average up-time of the ARENA server during one week of testing is 95%.

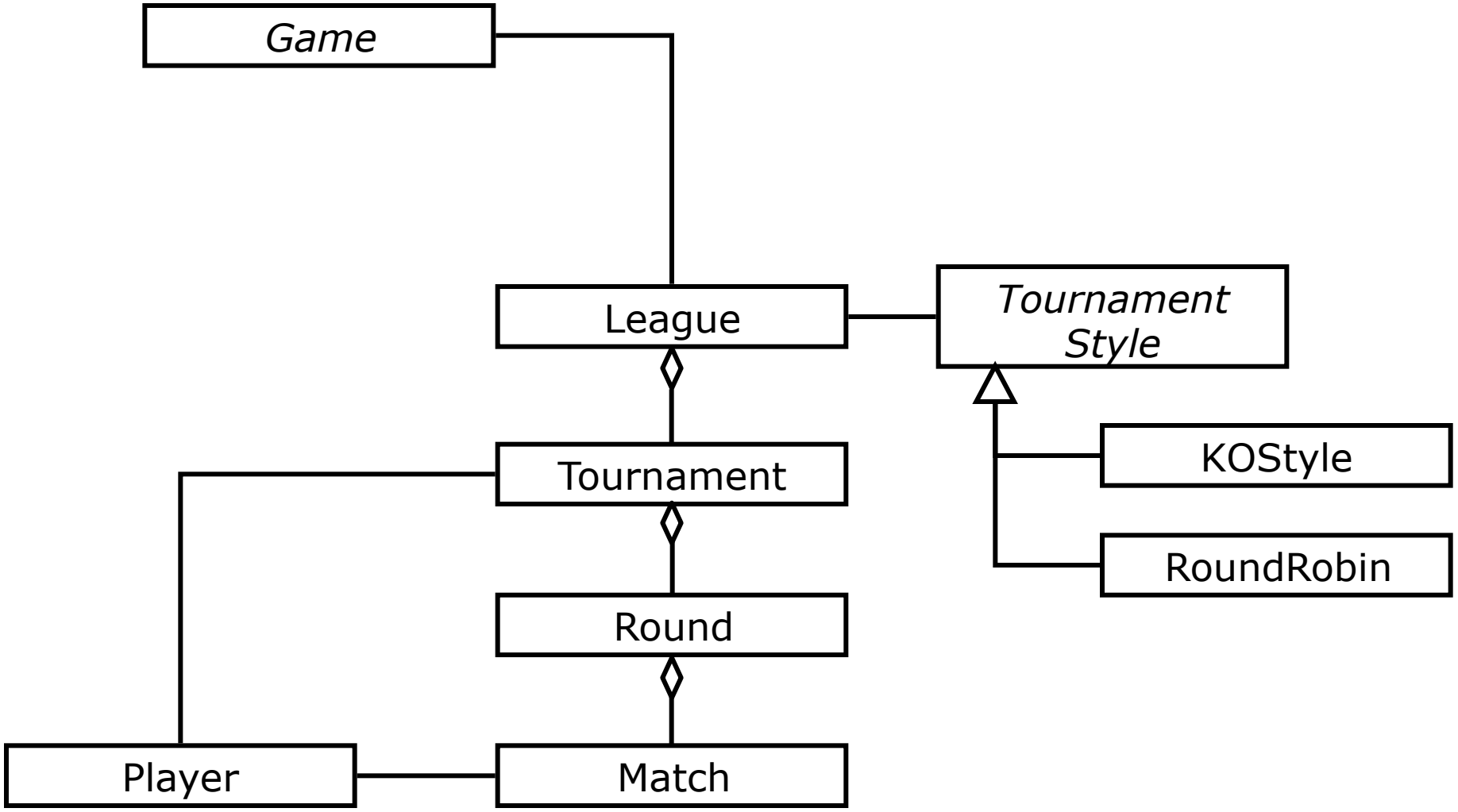
(Initial) ARENA Models

- Subsystem Decomposition
- User Interface of Client
- User Interface of Server
- Object Model

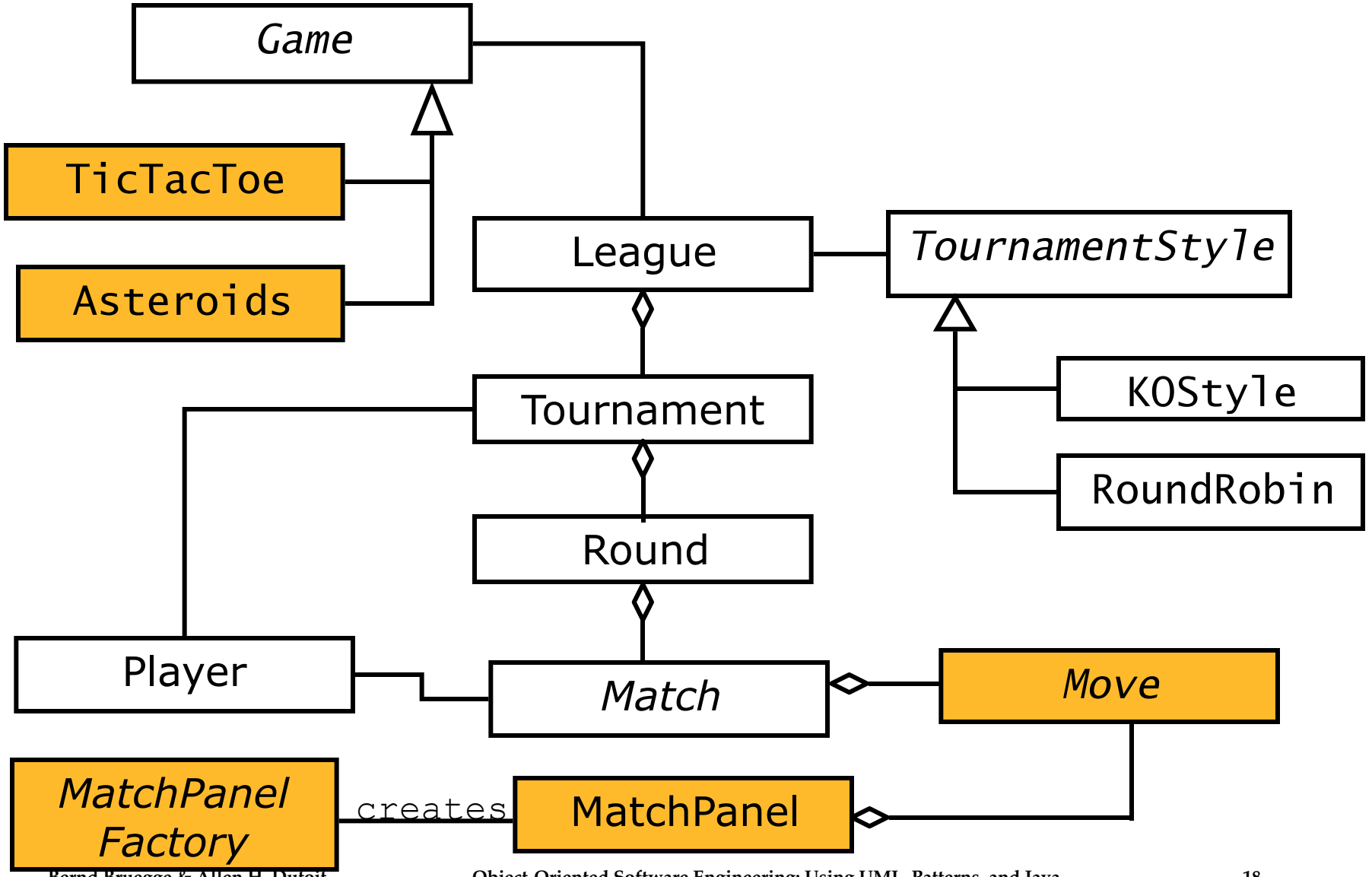
ARENA Subsystem Decomposition



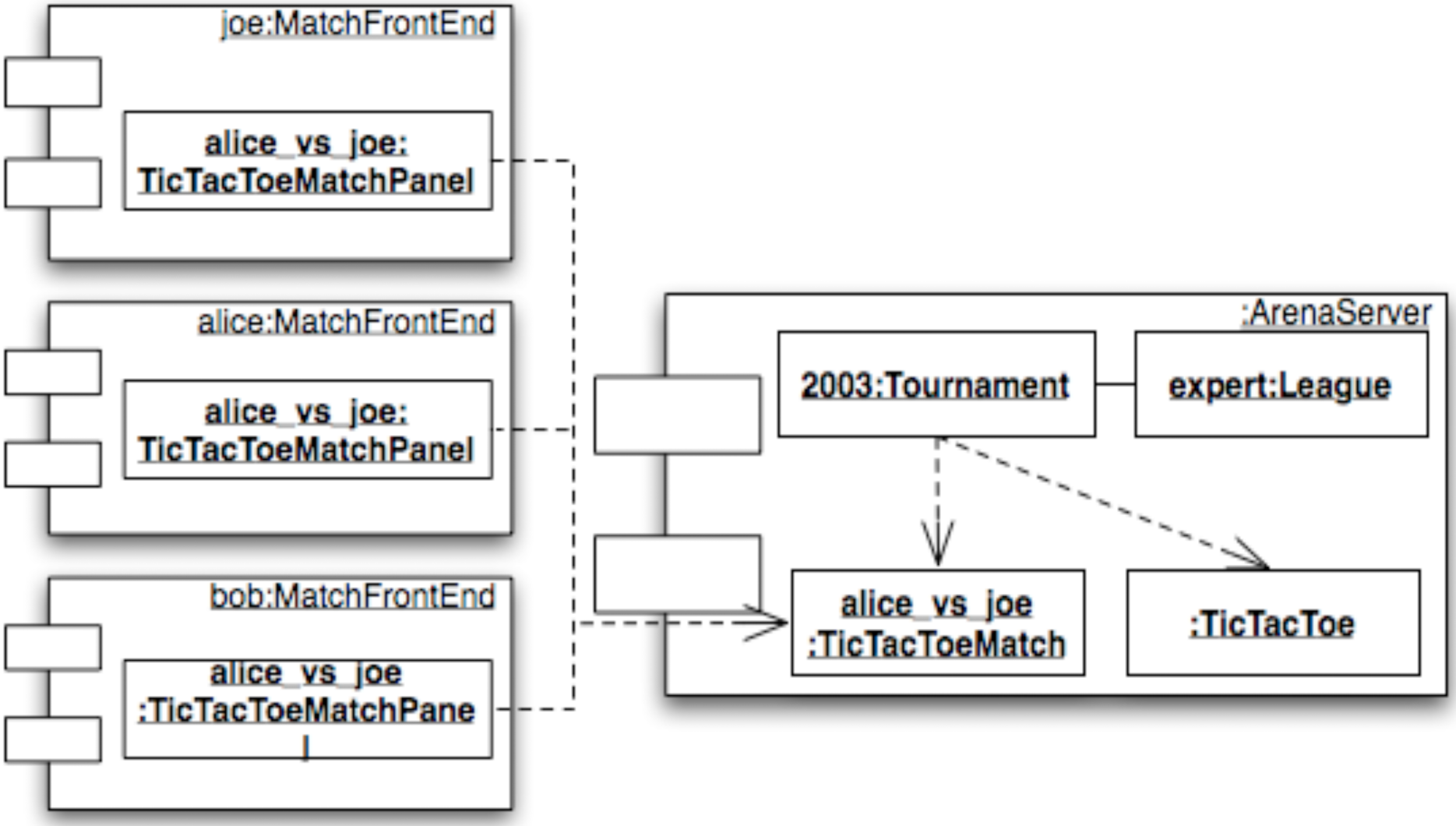
ARENA Object Model



ARENA Object Model (2)

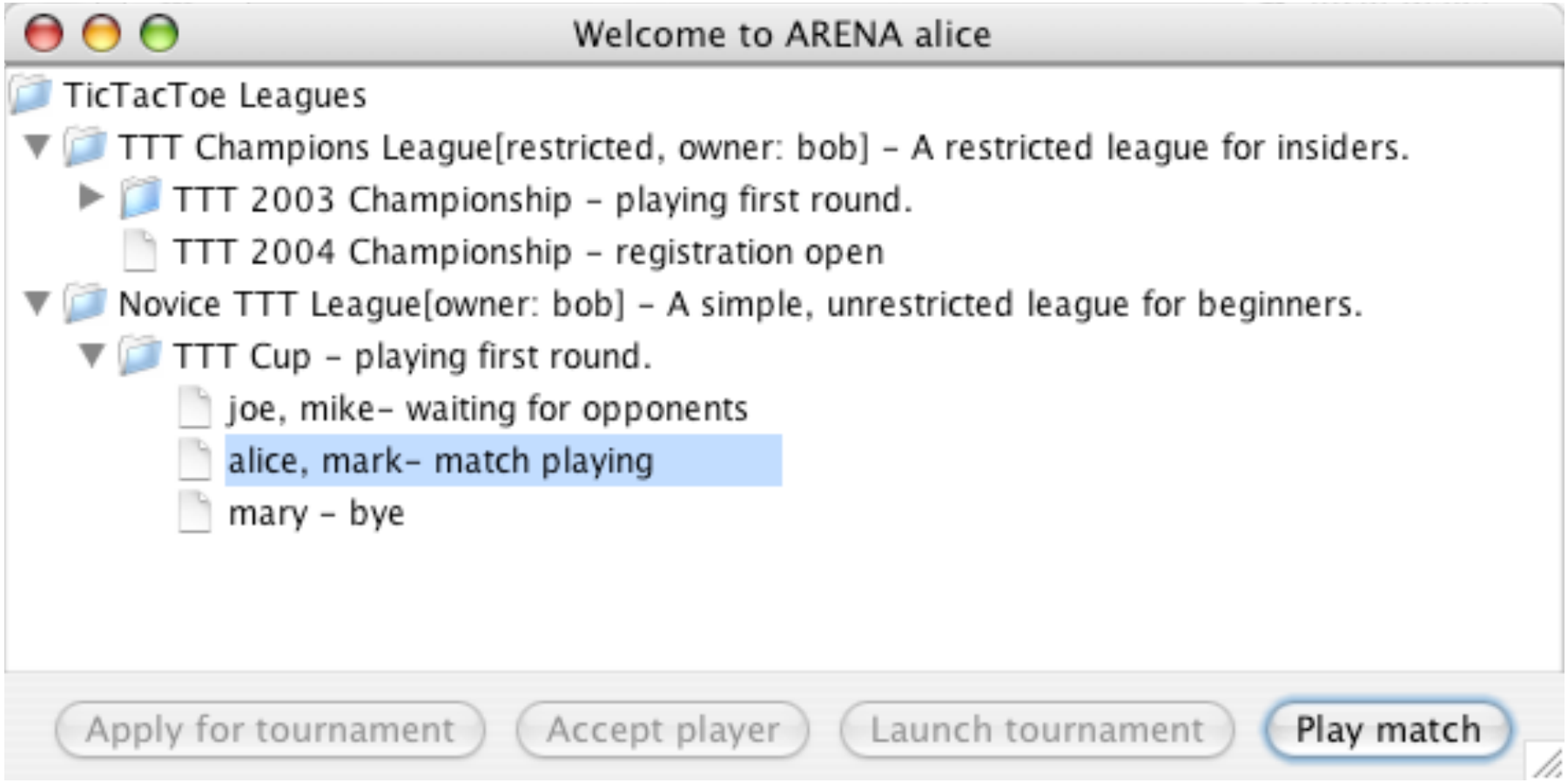


ARENA Instance-Diagram



ARENA User Interface (Client)

ARENA User Interface (Server)



More Information on ARENA

- *The ARENA Website:* *http://sysiphus.in.tum.de/arena*
- The ARENA case study is described at the end of each chapter, starting with Chapter 4
- Read Chapter 4.6