



الجامعة السورية الخاصة  
SYRIAN PRIVATE UNIVERSITY

المحاضرة الأولى

كلية الهندسة المعلوماتية

مقرر بنیان البرمجيات

# تذكرة بالمفاهيم الأساسية في هندسة البرمجيات

## المراحل الأساسية – لغة النمذجة UML

د. رياض سنبل

# Case Study: ARENA

---

## Description

- ARENA is a multi-user, Web-based system for organizing and conducting tournaments.
- ARENA is game independent in the sense that organizers can adapt a new game to the ARENA game interface, upload it to the ARENA server, and immediately announce and conduct tournaments with players and spectators located anywhere on the Internet.
- Organizers can also define new tournament styles, describing how players are mapped to a set of matches and how to compute an overall ranking of players by adding up their victories and losses (hence, figuring out who won the tournament).
- To recoup their operational costs, organizers can also invite potential sponsors to display advertisement banners during games.

# Requirements Elicitation & Initial Analysis

---



# How to Start a New Software Project!!

## Description

Web-based system for organizing and conducting

main terms, entities, and concepts (glossary)?

ent in the sense that org  
game to the ARENA game interface, upload it to the ARENA server, and immediately announce and conduct tournaments with players and spectators located anywhere on the Internet.

- Organizers can also define new tournament rules, how players are mapped to a set of matches and how to rank players by adding up their victories and losses (hence, figuring out who won the tournament)

Main actions, services, (use cases)

Main processes, flows, etc

- To recoup costs, organizers can also invite potential sponsors to display advertisement banners during games.

## Good Starting Point

- ARENA is a multi-user, Web-based system for organizing and conducting tournaments.
- ARENA is game independent. Organizers can adapt a new game to the ARENA game interface, upload it to the ARENA server, and immediately announce and conduct tournaments with players and spectators located anywhere.
- Organizers can also define new tournament styles, describing how players are mapped to a set of matches and how to compute an overall ranking of players by adding up their victories and losses (hence, figuring out who won the tournament).
- To recoup their operational costs, organizers can also invite potential sponsors to display advertisement banners during games.

Noun Phrases ≈ main concepts

Verbs ≈ Actions

## Descriptor

You need more clarifications  
from your client to understand  
some terms

banners

- ARENA is a **multi-user, Web** organizing and conducting
- ARENA is **game** independent. **organizers** can adapt a new **game interface**, upload it to the ARENA **server**, and immediately announce and conduct **tournaments** with **players** and **spectators** located anywhere on the Internet.
- **Organizers** can also define new **tournament styles**, describing how **players** are mapped to a set of **matches** and how to compute an overall **ranking of players** by adding up their victories and losses (hence, figuring out who won **the tournament**).
- To recoup their **operational costs**, **organizers** can also invite **potential sponsors** to display **advertisement banners** during **games**.

Matches

multi-user web-based system

operational costs

Organizers

Players

potential sponsors

ranking of players

Server

Spectators

the tournament

tournament styles

# New clarifications

→ Better Understanding

→ New possible Terms, actions, processes

■ ARENA is a multi-user system for organizing and conducting

■ ARENA is game interface where organizers can add new games, upload it to the ARENA server, and immediately conduct tournaments anywhere on the internet.

■ Organizers can describe how matches and of players by adding a tournament (hence, figuring out who won the tournament).

*A Tournament is a series of Matches among a set of Players. Tournaments end with a single winner. The way Players accumulate points and Matches are scheduled is dictated by the League in which the Tournament is organized.*

*The TournamentStyle defines the number of Matches and their sequence for a given set of Players. For example, Players face all other Players in the Tournament exactly once in a round robin TournamentStyle.*

Matches

multi-user web-based system

operational costs

Organizers

Players

potential sponsors

ranking of players

Server

Spectators

the tournament

tournament styles



## Description

- ARENA is a **multi-user, Web-based system** for **organizing and conducting tournaments**
- ARENA is **game independent** in the sense that **organizers can adapt a new game to the game interface, upload it to the ARENA, and immediately conduct tournaments** anywhere on the Internet
- **Organizers** can describe how **matches and of players** by adding a **League** (hence, figuring out who won **the tournament**)

*A League represents a community for running Tournaments. A League is associated with a specific Game and TournamentStyle. Players registered with the League accumulate points according to the **ExpertRating** defined in the League. For example, a novice chess League has a different ExpertRating formula than an expert League.*

*A Tournament is a series of Matches among a set of Players. Tournaments end with a single winner. The way players accumulate points and Matches are scheduled is dictated by **the League** in which the Tournament is organized.*

*The TournamentStyle defines the number of Matches and their sequence for a given set of Players. For example, Players face all other Players in the Tournament exactly once in a round robin TournamentStyle.*

operational costs  
Organizers  
Players  
potential sponsors  
ranking of players  
Server  
Spectators  
the tournament  
tournament styles

Finally!

**Table 4-4** Working glossary for ARENA. Keeping track of important terms and their definitions ensures consistency in the specification and ensures that developers use the language of the client.

<b>Game</b>	A Game is a competition among a number of Players that is conducted according to a set of rules. In ARENA, the term Game refers to a piece of software that enforces the set of rules, tracks the progress of each Player, and decides the winner. For example, tic tac toe and chess are Games.
<b>Match</b>	A Match is a contest between two or more Players following the rules of a Game. The outcome of a Match can be a single winner and a set of losers or a tie (in which there are no winners or losers). Some Games may disallow ties.
<b>Tournament</b>	A Tournament is a series of Matches among a set of Players. Tournaments end with a single winner. The way Players accumulate points and Matches are scheduled is dictated by the League in which the Tournament is organized.
<b>League</b>	A League represents a community for running Tournaments. A League is associated with a specific Game and TournamentStyle. Players registered with the League accumulate points according to the ExpertRating defined in the League. For example, a novice chess League has a different ExpertRating formula than an expert League.
<b>TournamentStyle</b>	The TournamentStyle defines the number of Matches and their sequence for a given set of Players. For example, Players face all other Players in the Tournament exactly once in a round robin TournamentStyle.

## Another Good Starting Point develop an example scenario

- ARENA is a **multi-user** organizing and conducting system.
- ARENA is **game** independent. **organizers** can adapt the **game interface**, upload it to the ARENA **server**, and immediately announce and conduct **tournaments** with **players** and **spectators** located anywhere on the Internet.
- **Organizers** can also define new **tournament styles** describing how **players** are mapped to a set of **matches** and how to compute an overall **ranking of players** by adding up their victories and losses (hence, figuring out who won **the tournament**).
- To recoup their **operational costs**, **organizers** can also invite **potential sponsors** to display **advertisement banners** during **games**.

Matches

multi-user web-based system

operational costs

### *Suggested Scenario*

*What if we want to launch a Chess Tournament? we understand better the client's expectation of the system, including the boundary of the system and the kinds of interactions between the user and the system*

tournament styles

# Requirements and High-Level Use Case Diagram

---

## Description

- ARENA is a **multi-user, Web-based system** for organizing and conducting tournaments.
- ARENA is **game** independent in the sense that **organizers** can adapt a new **game** to the ARENA **game interface**, upload it to the ARENA **server**, and immediately announce and conduct **tournaments** with **players** and **spectators** located anywhere on the Internet.
- **Organizers** can also define new **tournament styles**, describing how **players** are mapped to a set of **matches** and how to compute an overall **ranking of players** by adding up their victories and losses (hence, figuring out who won **the tournament**).
- To recoup their **operational costs**, **organizers** can also invite **potential sponsors** to display **advertisement banners** during **games**.

advertisement banners

Game

Game interface

Matches

multi-user

operational costs

Organizers

Players

potential sponsors

ranking of players


Server

Spectators

the tournament

tournament styles

League



Actors:  
Noun Phrases  
Again!

## Description

- ARENA is a **multi-user, Web-based system** for organizing and conducting tournaments.
- ARENA is **game** independent in the sense that **organizers** can adapt a new **game** to the ARENA **game interface**, upload it to the ARENA **server**, and immediately announce and conduct **tournaments** with **players** and **spectators** located anywhere on the Internet.
- **Organizers** can also define new **tournament styles**, describing how **players** are mapped to a set of **matches** and how to compute an overall **ranking of players** by adding up their victories and losses (hence, figuring out who won **the tournament**).
- To recoup their **operational costs**, **organizers** can also invite **potential sponsors** to display **advertisement banners** during **games**.

~~advertisement banners~~

~~Game~~

~~Game interface~~

~~Matches~~

Anonymous

~~multi-user~~

Admin

~~operational costs~~

~~Organizers~~

League owner

~~Players~~

Player

~~potential sponsors~~

advertiser

~~ranking of players~~

~~Server~~

~~Spectators~~

Spectators

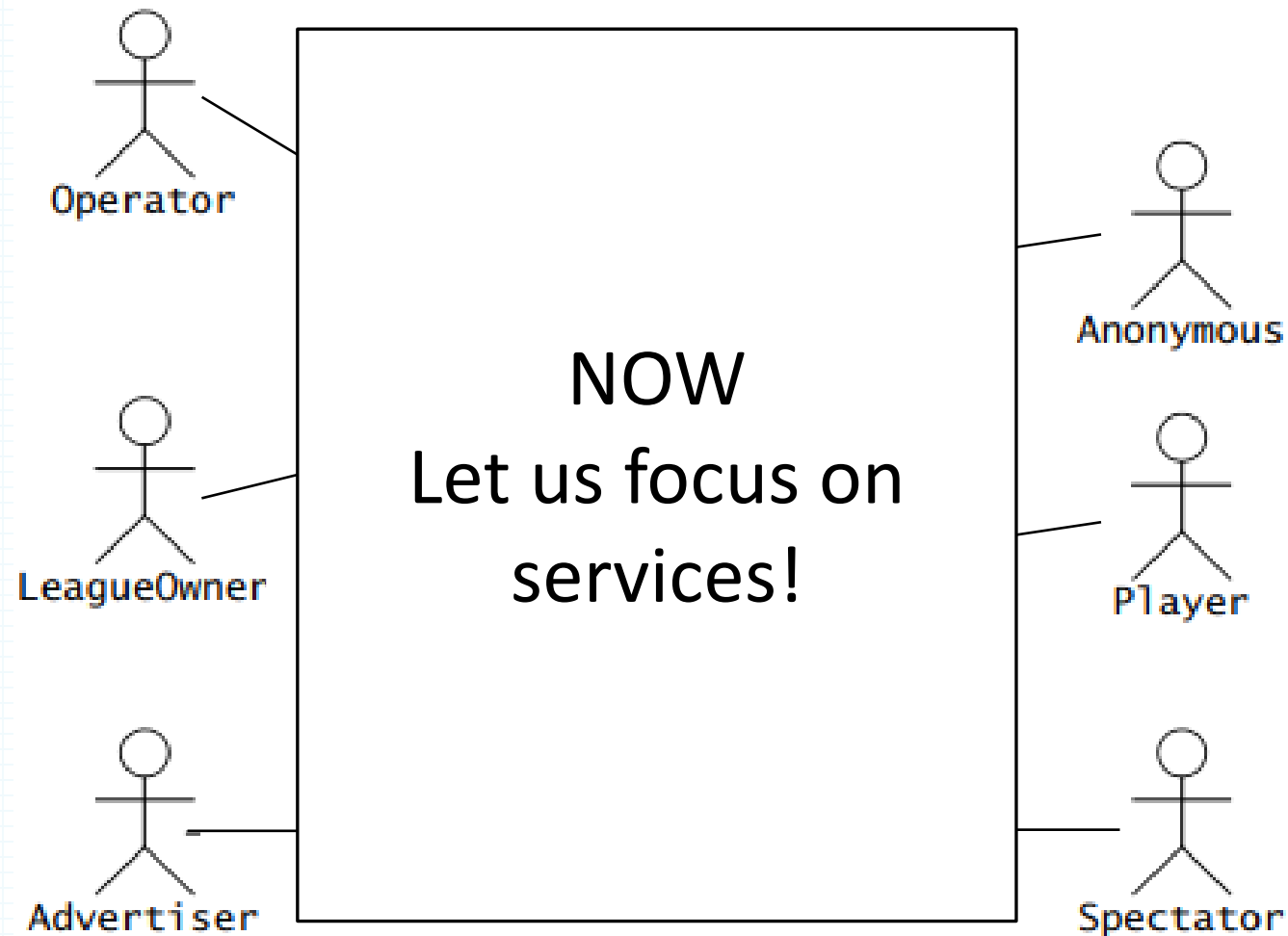
~~the tournament~~

~~tournament styles~~

~~League~~

# Till Now 😊

---



## Description

- ARENA is a multi-user, Web-based system for organizing and conducting tournaments.
- ARENA is game independent in the sense that organizers can adapt a new game to the ARENA game interface, upload it to the ARENA server, and immediately announce and conduct tournaments with players and spectators located anywhere on the Internet.
- Organizers can also define new tournament styles, describing how players are mapped to a set of matches and how to compute an overall ranking of players by adding up their victories and losses (hence, figuring out who won the tournament).
- To recoup their operational costs, organizers can also invite potential sponsors to display advertisement banners during games.



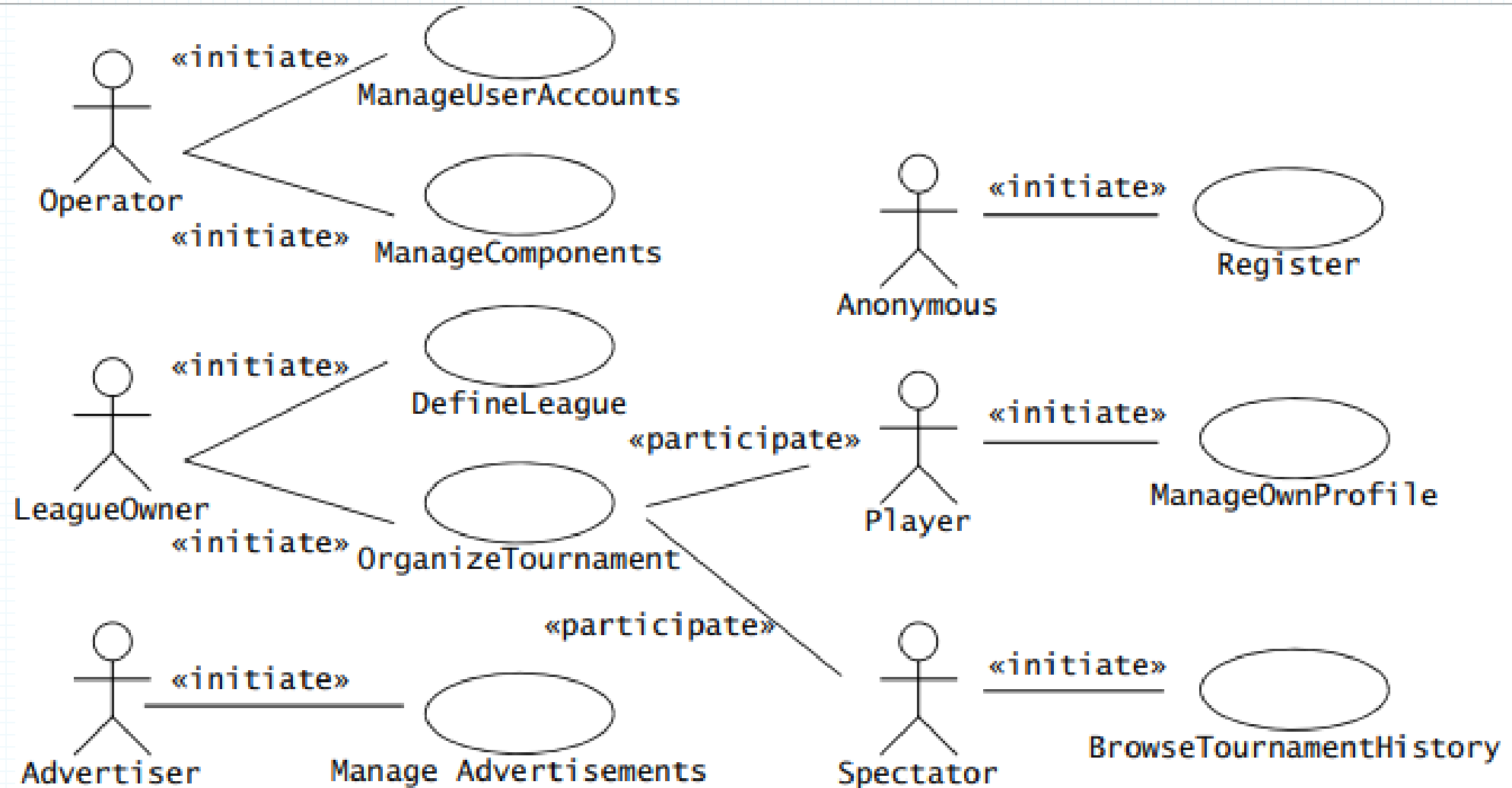


# Main services for each Actor

High Level  
Functional  
Requirements

- The **operator** should be able to **define** new games, **define** new tournament styles (e.g., knock-out tournaments, championships, best of series), **define** new expert rating formulas, and **manage** users.
- **League owners** should be able to **define** a new league, **organize** and announce new tournaments within a league, **conduct** a tournament, and **declare** a winner.
- **Players** should be able to **register** in an arena, **apply** for a league, **play** the matches that are assigned to the player, or **drop** out of the tournament.
- **Spectators** should be able to **monitor** any match in progress and **check** scores and statistics of past matches and players. Spectators do not need to register in an arena.
- **The advertiser** should be able to **upload** new advertisements, **select** an advertisement scheme (e.g., tournament sponsor, league sponsor), **check** balance due, and **cancel** advertisements.

# High-level use cases



# High-level use cases

<b>Register</b>	Anonymous users register with an Arena for a Player or a League-Owner account. User accounts are required before applying for a tournament or organizing a league. Spectators do not need accounts.
<b>ManageUserAccounts</b>	The Operator accepts registrations from LeagueOwners and for Players, cancels existing accounts, and interacts with users about extending their accounts.
<b>ManageComponents</b>	The Operator installs new games and defines new tournament styles
<b>DefineLeague</b>	The LeagueOwner defines a new league (
<b>OrganizeTournament</b>	The LeagueOwner creates and announces a new tournament, accepts player applications, schedules matches, and kicks off the tournament. During the tournament, players play matches and spectators follow matches. At the end of the tournament, players are credited with points
<b>ManageAdvertisements</b>	The Advertiser uploads banners and sponsors league or tournaments
<b>ManageOwnProfile</b>	The Players manage their subscriptions to mailing lists and answer a marketing survey.
<b>BrowseTournamentHistory</b>	Spectators examine tournament statistics and player statistics, and replay matches that have already been concluded (

# Non-Functional Requirements

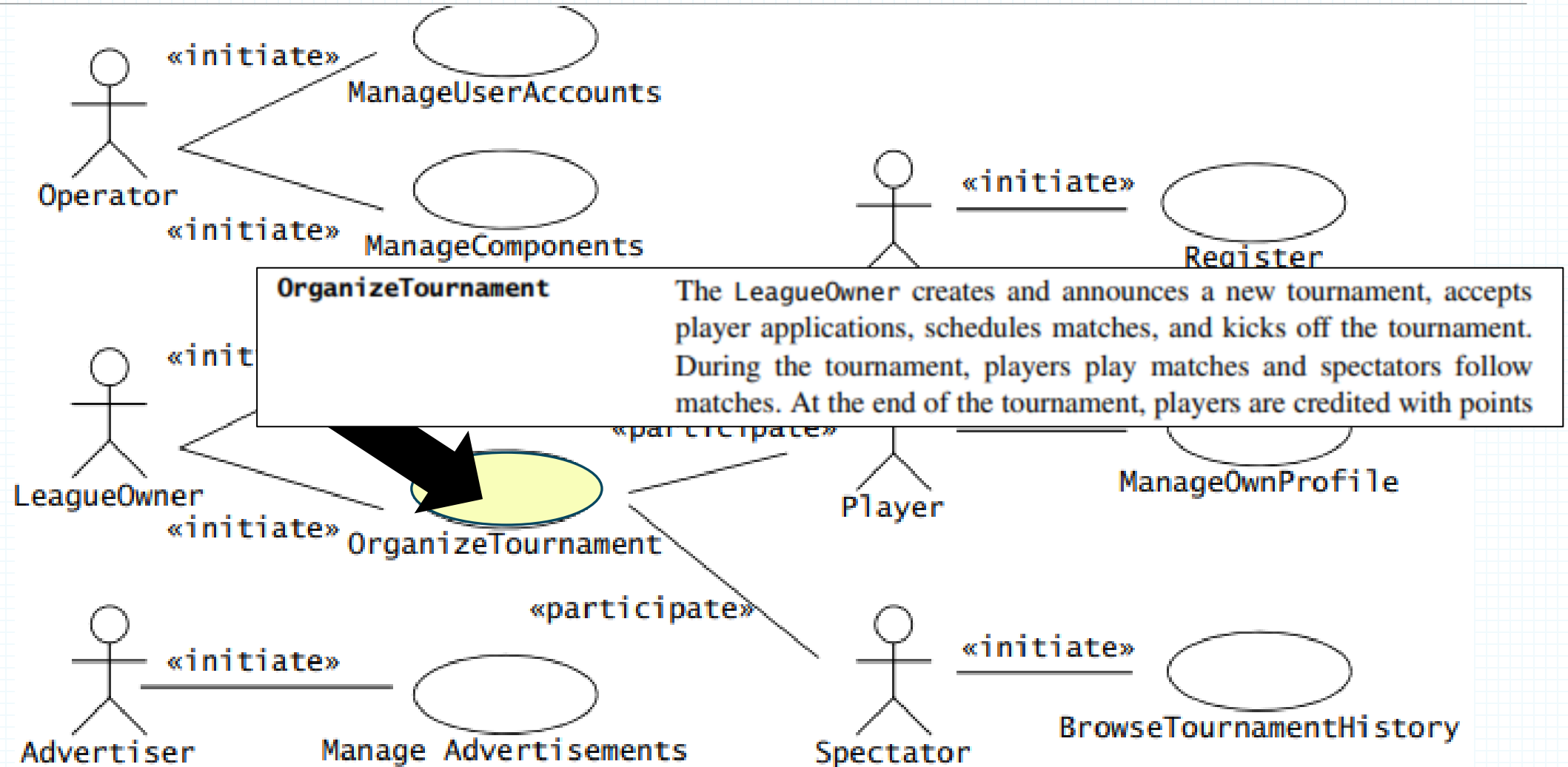
---

- **Low operating cost:** The operator must be able to install and administer an arena **without purchasing additional software** components and without the help of a full-time system administrator.
- **Extensibility:** The operator must be able to **add new games**, new tournament styles, and new expert rating formulas. Such additions may require the system to be temporarily shut down and new modules (e.g., Java classes) to be added to the system. However, no modifications of the existing system should be required.
- **Scalability:** The system must support the kick-off of **many parallel tournaments** (e.g., 10), each involving up to 64 players and several hundreds of simultaneous spectators.
- **Low-bandwidth network:** Players should be able to play matches via a 56K analog modem or faster.

# Detailed Use Cases

---

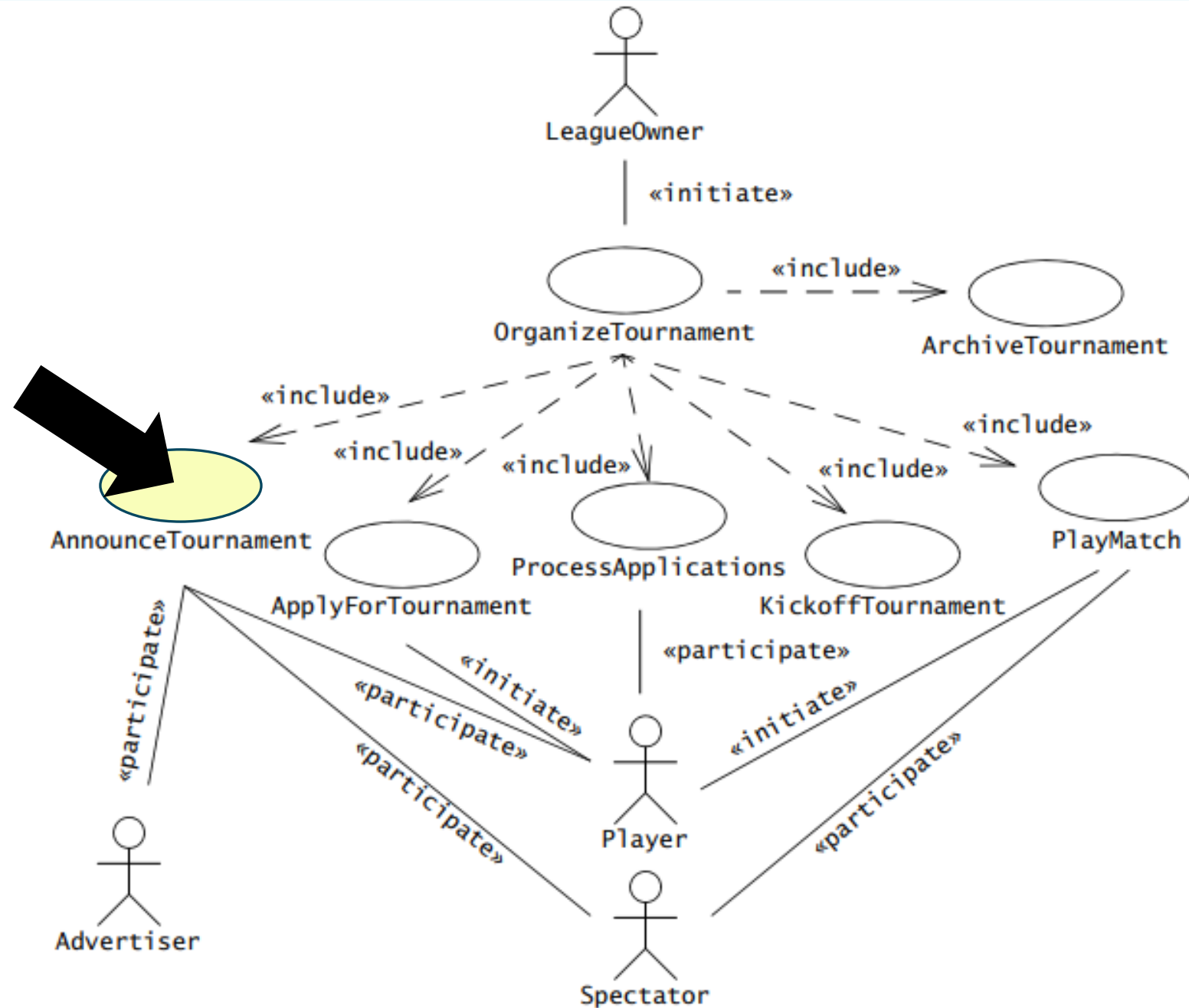
# High-level use cases



# Use Case Specification

<i>Use case name</i>	OrganizeTournament
<i>Participating actors</i>	Initiated by LeagueOwner Communicates with Advertiser, Player, and Spectator
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The LeagueOwner creates a Tournament, solicits sponsorships from Advertisers, and announces the Tournament (include use case AnnounceTournament).</li> <li>2. The Players apply for the Tournament (include use case ApplyForTournament).</li> <li>3. The LeagueOwner processes the Player applications and assigns them to matches (include use case ProcessApplications).</li> <li>4. The LeagueOwner kicks off the Tournament (include use case KickoffTournament).</li> <li>5. The Players compete in the matches as scheduled and Spectators view the matches (include use case PlayMatch).</li> <li>6. The LeagueOwner declares the winner and archives the Tournament (include use case ArchiveTournament).</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• The LeagueOwner is logged into ARENA.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• The LeagueOwner archived a new tournament in the ARENA archive and the winner has accumulated new points in the league, OR</li> <li>• The LeagueOwner cancelled the tournament and the players' standing in the league is unchanged.</li> </ul>

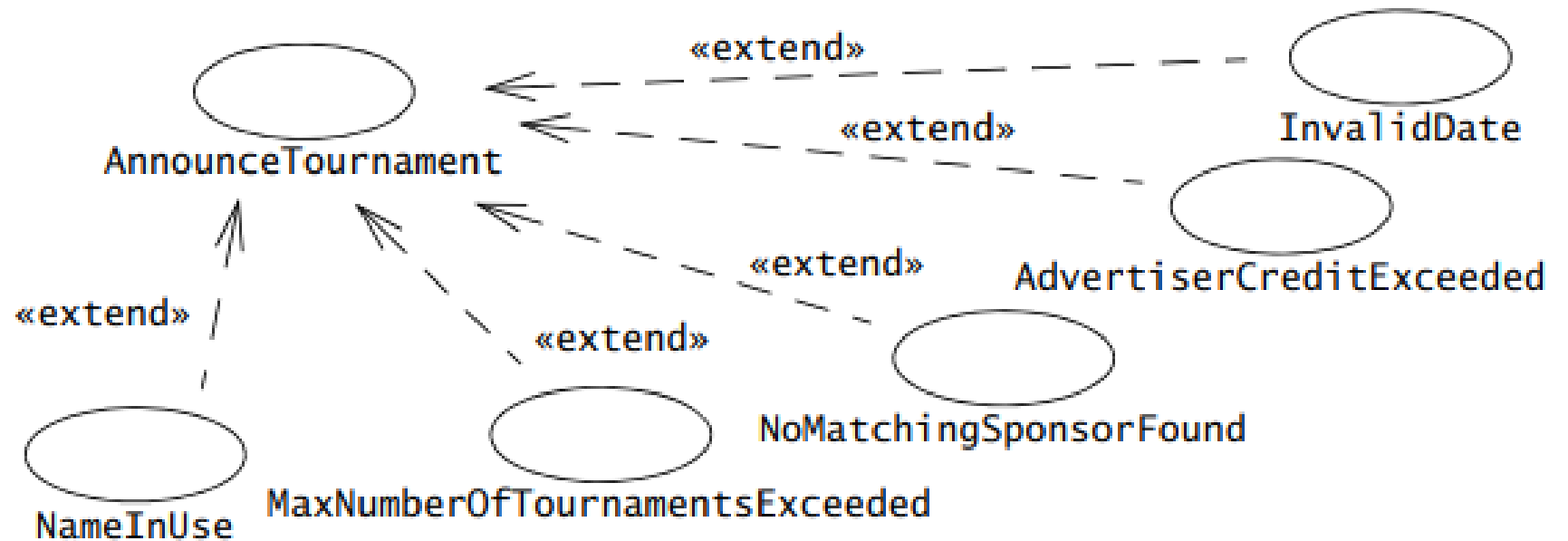
Detailed use cases  
refining the  
OrganizeTournament  
high-level use case.







# Deeper



## AdvertiserCreditExceeded

The system removes the Advertiser from the list of potential sponsors.

## InvalidDate

The system informs the LeagueOwner and prompts for a new date.

## MaxNumberOfTournamentsExceeded

The AnnounceTournament use case is terminated.

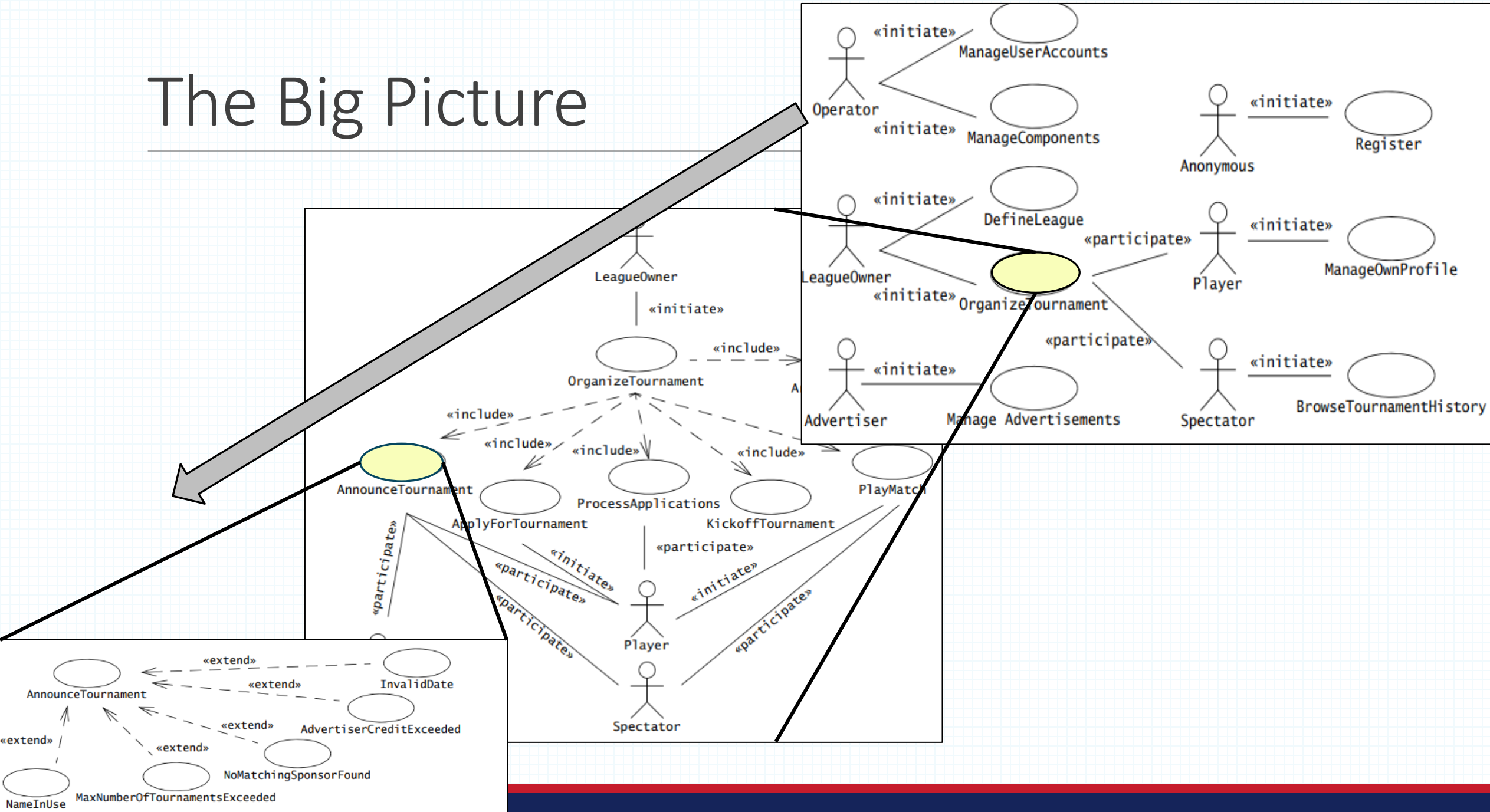
## NameInUse

The system informs the LeagueOwner and prompts for a new name.

## NoMatchingSponsorFound

The system skips the exclusive sponsor steps and chooses random advertisements from the advertisement pool.

# The Big Picture



# Identifying Entity Objects

---

a few simple heuristics  
to distinguish between  
noun phrases that  
correspond to objects,  
attributes, and  
associations

- Attributes represent a single property of an object.

*Flow of events*

1. The **LeagueOwner** requests the creation of a **tournament**.
2. The system checks if the **LeagueOwner** has exceeded the **number of tournaments** in the **league** or in the **arena**. If not, the system presents the **LeagueOwner** with a form.
3. The **LeagueOwner** specifies a **name, application start and end dates** during which **Players** can apply to the tournament, **start and end dates** for conducting the tournament, and a **maximum number of Players**.
4. The system asks the **LeagueOwner** whether an exclusive sponsorship should be sought and, if yes, presents a **list of Advertisers** who expressed the desire to be **exclusive sponsors**.
5. If the **LeagueOwner** decides to seek an exclusive sponsor, he selects a subset of the **names of the proposed sponsors**.
6. The system notifies the selected sponsors about the upcoming tournament and the **flat fee** for exclusive sponsorships.
7. The system communicates their **answers** to the **LeagueOwner**.
8. If there are interested sponsors, the **LeagueOwner** selects one of them.
9. The system records the **name** of the exclusive sponsor and charges the flat fee for sponsorships to the **Advertiser's account**. From now on, all **advertisement banners** associated with the tournament are provided by the exclusive sponsor only.
10. If no sponsors were selected (either because no **Advertisers** were interested or the **LeagueOwner** did not select any), the advertisement banners are selected at random and charged to each **Advertiser's** account on a per unit basis.
11. Once the sponsorship issues is closed, the system prompts the **LeagueOwner** with a **list of groups of Players, Spectators, and Advertisers** that could be interested in the new tournament.



a few simple heuristics  
to distinguish between  
noun phrases that  
correspond to objects,  
attributes, and  
associations

• Attributes have simple types.

• Complex concepts are  
represented as objects that  
are related to other objects  
with associations.

*Flow of events*

1. The **LeagueOwner** requests the creation of a **tournament**.
2. The system checks if the **LeagueOwner** has exceeded the **number of tournaments** in the **league** or in the **arena**. If not, the system presents the **LeagueOwner** with a form.
3. The **LeagueOwner** specifies a **name, application start and end dates** during which **Players** can apply to the tournament, **start and end dates** for conducting the tournament, and a **maximum number of Players**.
4. The system asks the **LeagueOwner** whether an exclusive sponsorship should be sought and, if yes, presents a **list of Advertisers** who expressed the desire to be **exclusive sponsors**.
5. If the **LeagueOwner** decides to seek an exclusive sponsor, he selects a subset of the **names of the proposed sponsors**.
6. The system notifies the selected sponsors about the upcoming tournament and the **flat fee** for exclusive sponsorships.
7. The system communicates their **answers** to the **LeagueOwner**.
8. If there are interested sponsors, the **LeagueOwner** selects one of them.
9. The system records the **name of the exclusive sponsor** and charges the flat fee for sponsorships to the **Advertiser's account**. From now on, all **advertisement banners** associated with the tournament are provided by the exclusive sponsor only.
10. If no sponsors were selected (either because no **Advertisers** were interested or the **LeagueOwner** did not select any), the advertisement banners are selected at random and charged to each **Advertiser's** account on a per unit basis.
11. Once the sponsorship issues is closed, the system prompts the **LeagueOwner** with a **list of groups of Players, Spectators, and Advertisers** that could be interested in the new tournament.

Entity Object	Attributes & Associations	Definition
<b>Account</b>	<ul style="list-style-type: none"> <li>balance</li> <li>history of charges (?)</li> <li>history of payments (?)</li> </ul>	An Account represents the amount currently owed by an Advertiser, a history of charges, and payments.
<b>Advertiser</b>	<ul style="list-style-type: none"> <li>name</li> <li>leagues of interest for exclusive sponsorships (?)</li> <li>sponsored tournaments</li> <li>account</li> </ul>	Actor interested in displaying advertisement banners during the Matches.
<b>Advertisement</b>	<ul style="list-style-type: none"> <li>associated game (?)</li> </ul>	Image provided by an Advertiser for display during matches.
<b>Arena</b>	<ul style="list-style-type: none"> <li>max number of tournaments</li> <li>flat fee for sponsorships (?)</li> <li>leagues (<i>implied</i>)</li> <li>interest groups (<i>implied</i>)</li> </ul>	An instantiation of the ARENA system.
<b>Game</b>		A Game is a competition among a number of Players that is conducted according to a set of rules. In ARENA, the term Game refers to a piece of software that enforces the set of rules, tracks the progress of each Player, and decides the winner.
<b>InterestGroup</b>	<ul style="list-style-type: none"> <li>list of players, spectators, or advertisers</li> <li>games and leagues of interests (<i>implied</i>)</li> </ul>	InterestGroups are lists of users in the ARENA which share an interest (e.g, for a game or a league). InterestGroups are used as mailing lists for notifying potential actors of new events.
<b>League</b>	<ul style="list-style-type: none"> <li>max number of tournament</li> <li>game</li> </ul>	A League represents a community for running Tournaments. A League is associated with a specific Game and TournamentStyle. Players registered with the League accumulate points according to the ExpertRating of the League.

Entity Object	Attributes & Associations	Definition
<b>LeagueOwner</b>	<ul style="list-style-type: none"> <li>name (<i>implied</i>)</li> </ul>	The actor creating a League and responsible for organizing Tournaments within the League.
<b>Match</b>	<ul style="list-style-type: none"> <li>tournament</li> <li>players</li> </ul>	A Match is a contest between two or more Players within the scope of a Game. The outcome of a Match can be a single winner and a set of losers or a tie (in which there are no winners or losers). Some TournamentStyles may disallow ties.
<b>Player</b>	<ul style="list-style-type: none"> <li>name (<i>implied</i>)</li> </ul>	
<b>Tournament</b>	<ul style="list-style-type: none"> <li>name</li> <li>application start date</li> <li>application end date</li> <li>play start date</li> <li>play end date</li> <li>max number of players</li> <li>exclusive sponsor</li> </ul>	A Tournament is a series of Matches among a set of Players. Tournaments end with a single winner. The way Players accumulate points and Matches are scheduled is dictated by the League in which the Tournament is organized.

# Identifying Boundary Objects

- Boundary objects represent the interface between the system and the actors. They are identified from the use cases and usually represent the user interface at a coarse level.

Boundary Object	Definition
<b>TournamentForm</b>	Form used by the LeagueOwner to specify the properties of a Tournament during creation or editing.
<b>RequestSponsorshipForm</b>	Form used by the LeagueOwner to request sponsorships from interested Advertisers.
<b>SponsorshipRequest</b>	Notice received by Advertisers requesting sponsorship.
<b>SponsorshipReply</b>	Notice received by LeagueOwner indicating whether an Advertiser wants the exclusive sponsorship of the tournament.
<b>SelectExclusiveSponsorForm</b>	Form used by the LeagueOwner to close the sponsorship issue.
<b>NotifyInterestGroupsForm</b>	Form used by the LeagueOwner to notify interested users.
<b>InterestGroupNotice</b>	Notice received by interested users about the creation of a new Tournament.



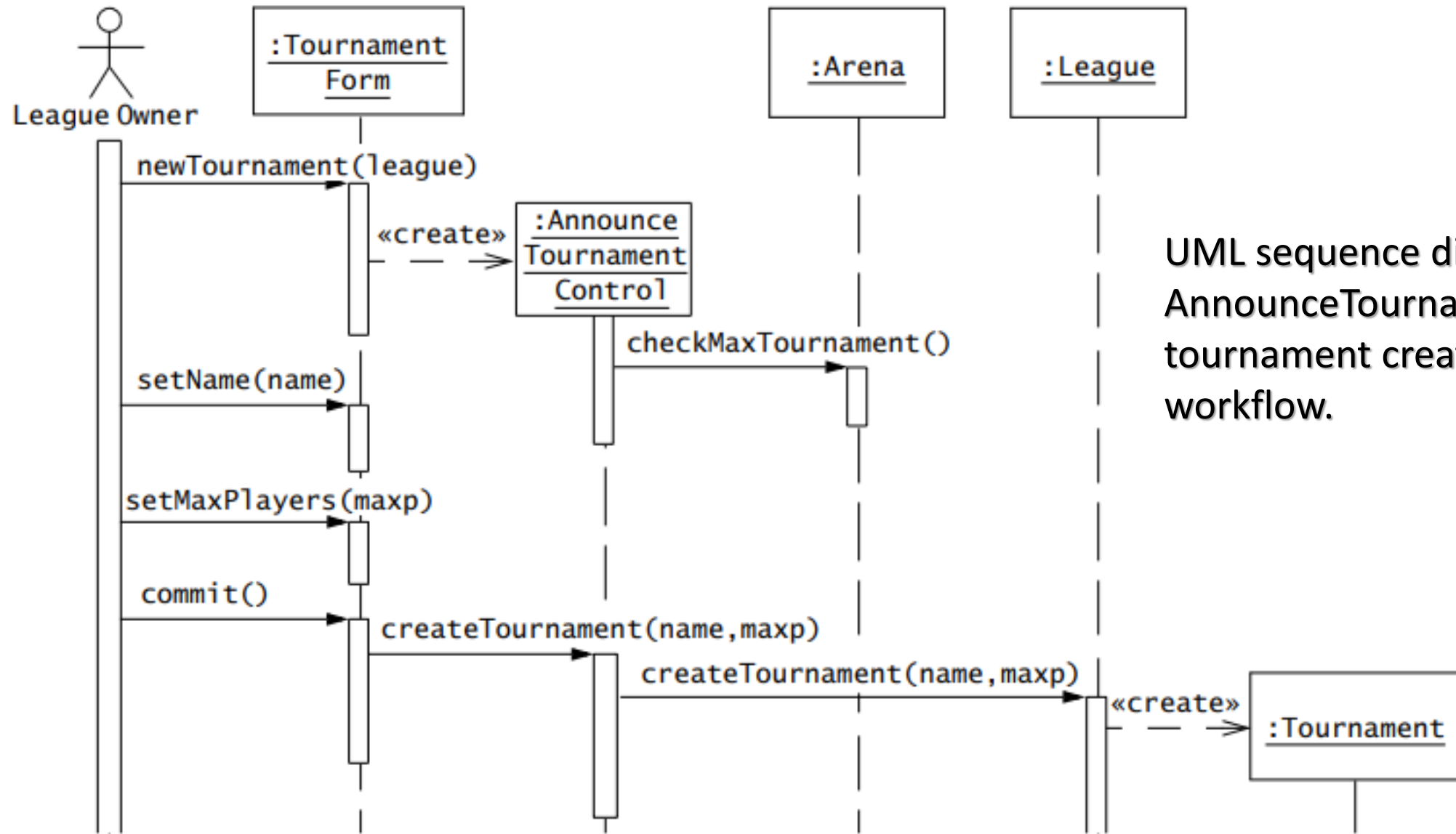
# Identifying Control Objects

---

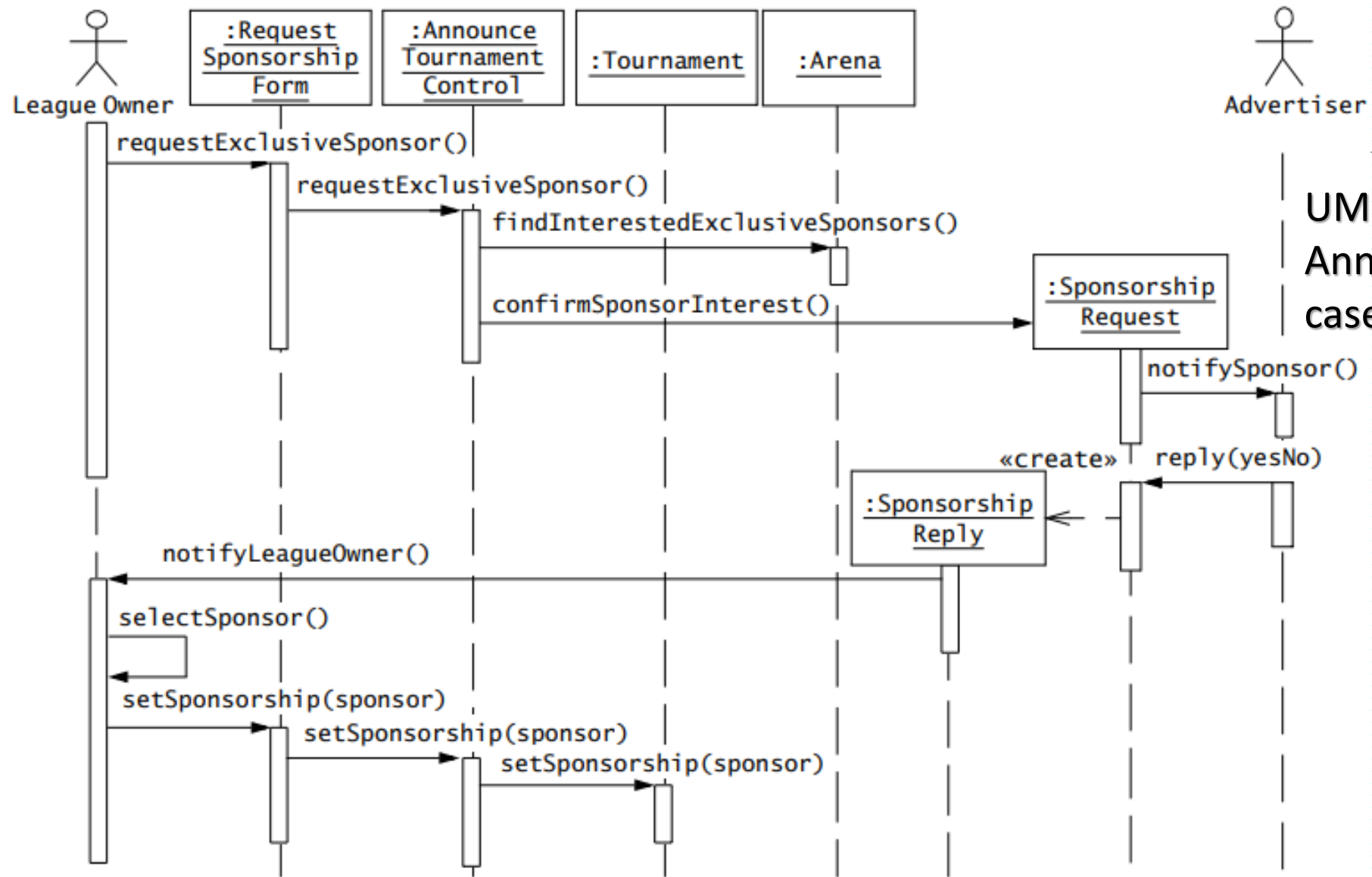
- Control objects represent the coordination among boundary and entity objects.
- In the common case, a single control object is created at the beginning of the use case and accumulates all the information needed to complete the use case, then destroyed with the completion of the use case
- Example:
  - **AnnounceTournamentControl**, which is responsible for sending and collecting notices to Advertisers, checking resource availability, and, finally, notifying interested users

# Modeling Interactions Among Objects

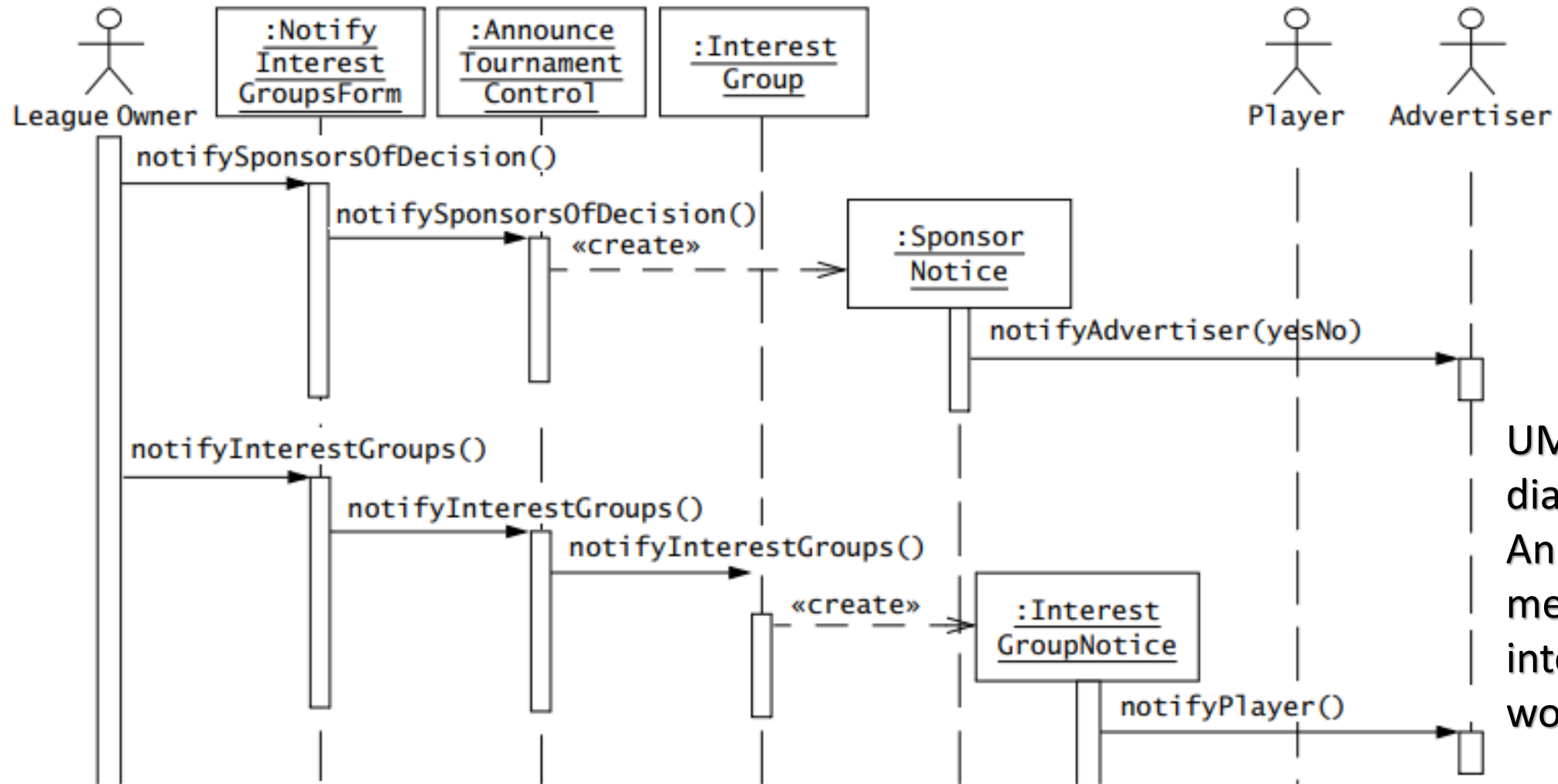
---



UML sequence diagram for AnnounceTournament, tournament creation workflow.

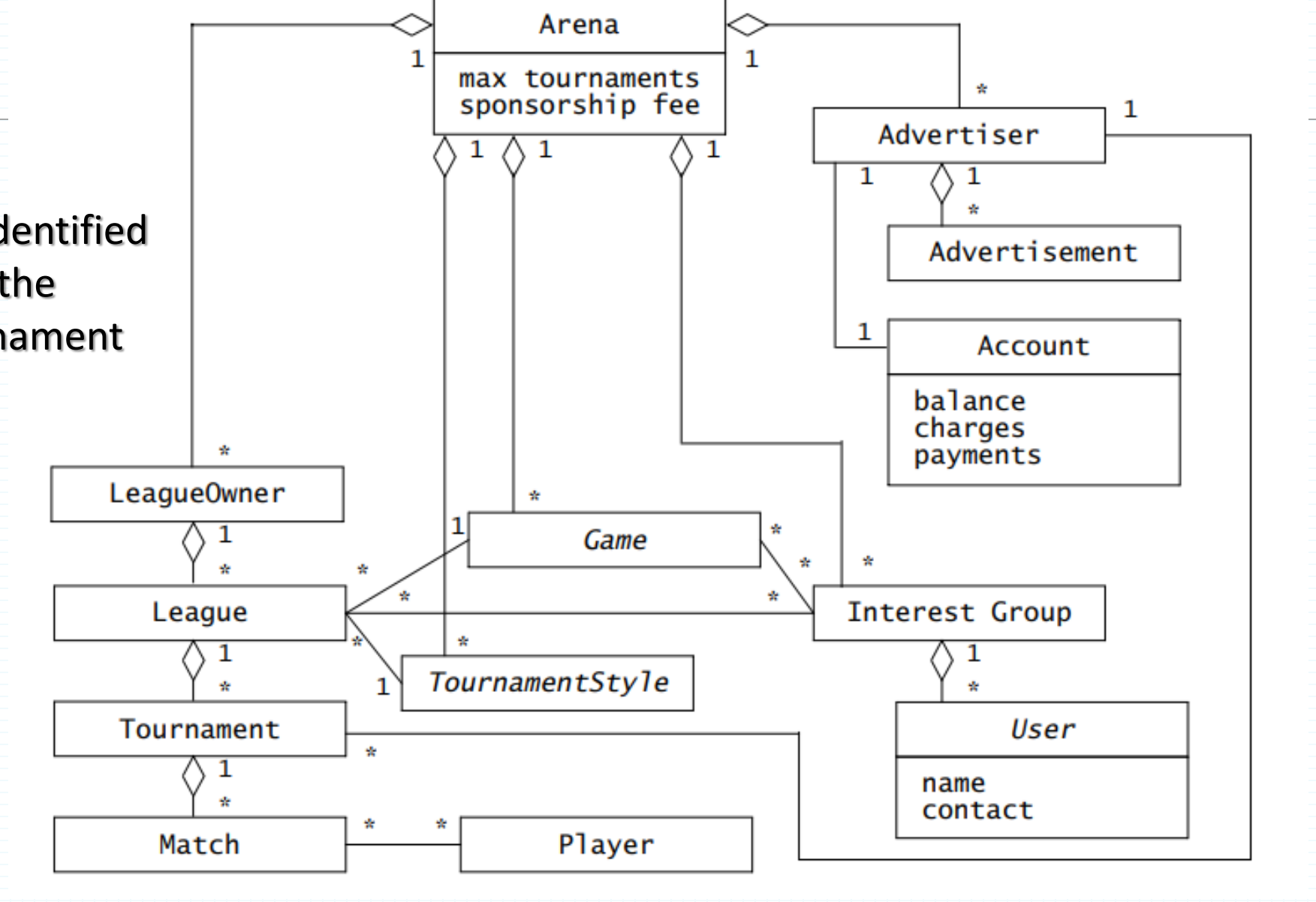


UML sequence diagram for AnnounceTournament use case, sponsorship workflow



UML sequence diagram for AnnounceTournament use case, interest group workflow

Entity objects identified after analyzing the AnnounceTournament use case



Inheritance  
hierarchy among  
entity objects of  
the  
AnnounceTournamen  
t use case

