Course: Software Development Methods.
Prof. Luca Dan Serbanati
Conf. Andrei Vasilateanu

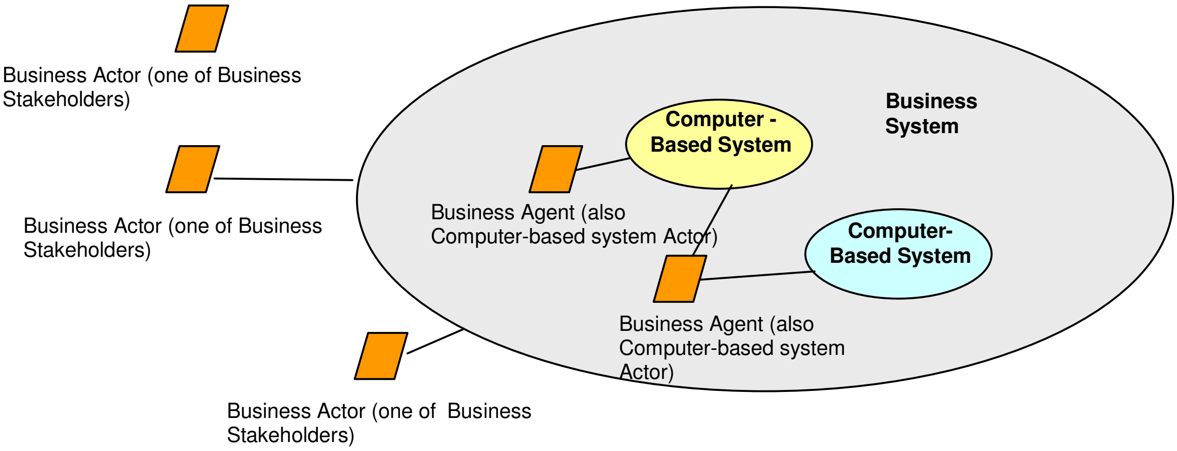
GUIDELINES FOR LABORATORY AND PROJECT ACTIVITY

SECOND MACRO-ACTIVITY: SOFTWARE USER REQUIREMENTS ANALYSIS

HW2. User Requirements Analysis

Deliveries: Context Diagram and Use Case Model

The following figure shows the relationships between the business stakeholders, the business system, and its internal computer-based systems.


Business Actor (one of Business Stakeholders)
Business Actor (one of Business Stakeholders)
Business Actor (one of Business Stakeholders)
Business System
Computer - Based System
Business Agent (also Computer-based system Actor)
Computer- Based System
Business Agent (also Computer-based system Actor)

1. Define the computer-based system boundary and construct its context diagram. In order to do this, follow the next steps:

   a. Write the system description consisting from only few phrases which provide a general, concise description of the system (the problem system).
   b. Identify the stakeholders in the business system who are interested in a computer-based system solution and are not developers. Some of them may directly interact with the computer-based system: they are actors of the computer-based system. Describe their objectives (goals) and interests or concerns regarding the system.
   c. According to the analysis of the stakeholders, delineate the problem domain and implicitly the boundary of the computer-based system.

2. Construct the Software Requirements Document as a **Use Case Model**[1]. In order to do this, follow the next steps:

[1] The Use Case Model is composed from a diagram illustrating the scope of the application being built and use case descriptions. The diagram contains actors (roles played by people or systems external to the

a. According to the analysis of the stakeholder goals, identify the primary actors.

b. During interaction a primary actor generates events to the system, usually requesting some operation in response. Identify the events the primary actors trigger for stimulating the computer-based system. Document the system events in the System Event List. It may contain three types of events: data flow, temporal and control. System events should be expressed at the level of intent rather than in terms of the physical input medium or interface widget level.

c. For each event in the System Event List identify a use case the system should implement. These use cases specify the system functions deduced from the stakeholders' concerns and system events. A function of the system should follow the following pattern:
"The system has to do/execute/produce/provide/memorize/store…"

d. Describe each identified use case with the use case template provided at the course. Warning: include both the main and alternative flows.

e. Add for each use case description the quality attributes you extract from the stakeholders' objectives: portability, robustness, usability, performances, software or hardware platform on which the system will have to execute, so on.

f. Draw the software use case diagram.

3. Construct the **System Sequence Diagram.**

   Use cases describe how external actors interact with the software system we are interested in. During this interaction an actor generates events to a system, usually requesting some response. To identify system operations it is necessary to have a clear choice of system events. It is desirable to isolate and illustrate the services that an external actor requests to a system: they are an important part of the system behavior understanding. UML includes **sequence diagrams** as a notation that illustrate interaction scenarios between actors and the system as well as interactions between objects in the system.

   A system sequence diagram is a picture that shows, for a particular scenario of a use case, the events that external actors generate and their ordering, and inter-system events. For this the system is treated as a black box; the emphasis of the diagram is put on events that cross the system boundary from actors to systems.

   a. Identify the system operations as first messages in the system sequence diagrams.

   b. Design system sequence diagrams for the main success scenario of the use case, and frequent or complex alternative scenarios.

4. Design **a contract for each system operation**. The contract should emphasize the operation post-conditions. These post-conditions should describe what is new in the software system state after the operation execution: new objects or links appeared, attribute values modifications, or objects or links eliminations.

   Hint: for describing post-conditions it is useful to know the problem resources as they will be described in the problem domain model in the next homework.

application being built) and uses cases that is the services or functions the actors request from the application.

Do not confuse Business Use Case Model with the Use Case model! A single Business Use Case Model may have many (system) Use Case Models associated with it, where each Use Case Model represents a single computer-based application.