CS 578 – Software Architectures
Spring 2009
Homework Assignment #2
Due: *Tuesday, March 3, 2009*
– see course websites for submission details –

During early development phases of a software system, the informal system requirements are often translated into more formal models to allow different types of early analyses and to support subsequent development of more concrete design artifacts. For example, natural language descriptions of the required behavior in particular use cases are commonly represented via sequence diagrams which depict the resulting interactions among the system's constituent components. Further, different states in the system execution are often captured using state variables, values of which can restrict the possible execution sequences (e.g., we can define pre- and post-conditions on these variables for different component operations).

Figure 1 depicts an example system specification diagram of a web caching system. The depicted UML sequence diagram corresponds to the following use case scenario: "When the Client requests a web page that is not cached, the page should be fetched from the Server and returned to the Client." Additionally, the "OCL-like[1]" constraints depicted in Figure 1 describe the conditions that should be satisfied in order to invoke a particular operation. For example, the `requestServerData` operation can be invoked only if the particular page is not cached.

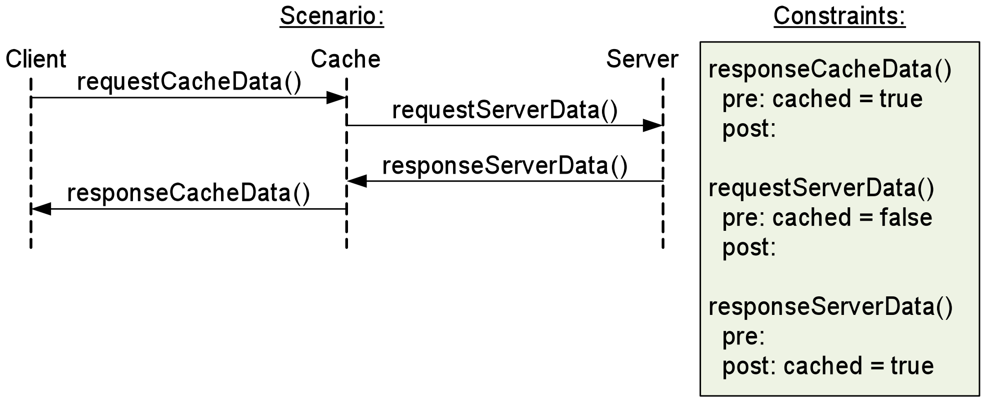

Figure 1. Sequence diagram and operation constraints for a simple web caching system.

In this assignment we provide a set of high-level natural language specifications of use cases of the C4 system from homework #1. These descriptions also contain hints suggesting possible system states. The provided specifications are reflective of the types of natural language specifications that are often given for real development projects.

[1] OCL is a commonly used constraint-modeling notation often used in conjunction with UML. For the purposes of this assignment, we are not asking you to adhere to OCL syntax, but rather use first order logic constraints like those in Figure 1.

Your task is to depict how one of the architectures you developed in homework #1 supports the provided set of C4 use cases. The execution sequences of the use cases should be captured as UML sequence diagrams similar to Figure 1. You are also required to define pre- and post-conditions for execution of operations occurring in your sequence diagrams. These conditions should be captured as OCL-like constraints on system state variables. You can extract the system state variables relevant for your system from the provided specifications (e.g., you can have a variable `requestPending`).

The use cases of the C4 system you need to model are as follows:

1. **Successful ISDN request submission.** An administrator can submit a request for activation of ISDN services on user's behalf. The user's request is accepted after checking with the database if ISDN option exists for the user's site, and the user has no other pending requests. The request is forwarded to NOSS subsystem.
2. **Unsuccessful ISDN request submission.** A user's request for ISDN services is rejected and the user is informed if the user's site does not have an ISDN option or the user has other pending requests.
3. **ISDN is physically installed.** The service provider can install ISDN support at the user's site. This scenario happens only when there is a pending user's ISDN request. In the system, this will be reflected with the exchange of events in which the NOSS system sends confirmation of successful installation to the C4 system.
4. **ISDN will not be physically installed.** The service provider can refuse to provide ISDN for a user if the management decides that it is not cost efficient after all. This scenario happens only when there is a pending user request. In the system, this will be reflected with the exchange of events in which the NOSS subsystem sends refusal message to C4. Subsequently, the ISDN option for the user is disabled.
5. **Administrator enables ISDN option.** A manager operating on one of the Downstream subsystem can enable ISDN option for a user. This can be done only if the user does not have that option already enabled.
6. **ISDN request completed.** An administrator can check on the completion status of a user's ISDN request. If the request is completed, the user is informed about the outcome.
7. **Successful phone plan request submission.** An administrator can submit a request for a change of phone plan on user's behalf. The user's request is accepted if the user has no other pending requests. The request is forwarded to the Billing subsystem.
8. **Unsuccessful phone plan request submission.** A user's request for ISDN services is rejected and the user is informed if she has other pending requests.
9. **Phone plan modified.** When the requested phone plan modification has been successfully completed, the Billing system reports it to the C4 system. The administrator subsequently informs the user about the successful completion of the request.
10. **Subscription to new business options.** A user can subscribe to receiving information about the published business options. This is first reported to the database. If the user is not already subscribed, the Downstream system in charge of publishing new business options is notified about the user's subscription. The Billing system is also notified.

11. **Unsubscription from new business options.** A user can unsubscribe from receiving information about the published business options. This is stored in the database. If the user has actually been subscribed (users without subscriptions can still make these requests), the Downstream system in charge of publishing new business options is notified about the cancellation of the user's subscription. The Billing system is also notified.
12. **New business options notification.** The Downstream system can publish new business options. If a user is subscribed, the C4 administrator will be notified about it in order to forward the information to the subscribed user.
13. **Disconnecting ISDN.** If a user did not pay the bills, the Billing system will notify the C4 system. The C4 system will then manage the necessary processing: the administrator will notify the user, the ISDN option will be disabled and the NOSS system will be requested to physically disconnect the user's ISDN line.

QUESTIONS

1) For each of the use cases above, create a UML sequence diagram for your C4 architecture complete with OCL-like constraints on system variables. If your architecture DOES NOT support a particular use case, do not change your architecture, but rather explain in detail why your architecture as designed in Homework 1 does not support the use case.
2) Briefly describe any discovered deficiencies with the given scenario specifications, discrepancies between different scenarios, or potential problems when executing different scenarios in your system (e.g., deadlock conditions). **Be specific: For example, note where your constraints have been violated.**
3) Having supported each of these use cases (or attempted to support each), how would you modify your C4 architecture to better address your scenarios. How would you resolve inconsistencies and simplify component interactions?

PLEASE NOTE: For the purposes of this assignment, you can focus your models on the case of a single user who is always being serviced through the same administrator. Additionally, an administrator can do only one thing at a particular time instant (e.g., an administrator cannot report a business offer at the same time she is trying to submit an ISDN request on the user's behalf). The given specifications are at a very high level of abstraction and provide a lot of room for interpretation, which will depend on your system's architecture. If you are making substantial assumptions besides the ones given above, please write these down.