OO COURSE OUTLINE

A Constructivism Based Approach

The OO Conceptual Model

Simplified versions of use cases, class diagrams, OIDs, scenario diagrams in a very informal UML notation, are used to create draft models to highlight the structure and behaviour of the System.
Concepts are presented without any reference to any programming language.

The Goody's Example

Students are already familiar with "Goody's", they have all used its services and have an understanding of its structure and behaviour.

Consider the worlds:

- Goody's
- Olga Square Goody's (OSG)

They are both identifiers of entities (objects).
The first one is used to identify a conceptual object (a type) that defines the structure and behaviour of Goody's restaurants.
The second one identifies a specific real-world object (an instance) that one located in Olga Square (see figure 1), whose structure and behaviour is specified by the first one.

Figure 1. The Olga Square Goody's

Students have a clear understanding of the concept of type and instance and they are only required to learn the terms used to refer to them. They all know that in order to eat a hamburger they must find an instance of "Goody's" and send a message to it.

They already know that it is preferable to use the service of an existing instance, if one is available, rather than build their own.

Ask students to identify classes and instances from a text that describes Goody's, as well as from other texts that describe the creation and operation of the OSG. They should be guided to represent them using the UML artifacts for class and instance.

System's structure
Students have a good understanding of the composition of this type of system (restaurant). However this understanding is not for the entire system (object) but only for the front-end of it, i.e., the part with which they have to interact (see figure 2).

Figure 2. Front-end of Olga Square Goody's

The back-end of "Goody's" (see figure 3) is actually hidden from its clients; only some components of the front-end can interact with some components of the back-end.

Students also understand the reasons for such packaging and controlled visibility. It is a well-organized restaurant that provides reliable services within time and budget. And this is the reason they prefer using this type of objects.

Describe the structure (composition) of Goody's restaurants.
Use a simplified UML notation of the Class diagram to represent this structure.
Introduce the concepts of Aggregation and Generalization/Specialization.

Figure 3. Back-end of Olga Square Goody's

System's behaviour

The system provides a set of services in response to messages that accept from the environment.
"objects do things and we ask them to perform what they do by sending them messages" G. Booch.

Consider Helen that has just accepted (see figure 4) the following message from Chris, which is a client. "a hamburger and a cheese pie, please".

Figure 4. Helen that has just accepted the order of Chris.

Describe the behaviour of Helen to this message.
A sample description follows:
"Helen sends an oral message to Nick (see figure 5) to prepare a hamburger, but she has to properly set the *time* button of the microwave oven and then press the *start* button to heat the cheese pie. Nick, in the back-end, uses the services of the toaster to prepare the hamburger. The dishwasher sends the message "need water" to the object tap, in a different way than it sends the message "washing finished" to Nick. …"

Figure 5. Helen sends an oral message to Nick to prepare a hamburger

[Optional] Use the concept of Use case to describe the systems behaviour.

Use an Object Interaction Diagram or scenario diagram to represent the Objects and the way they collaborate for the system to provide the required behaviour.
Students are firstly asked to describe the whole interaction in natural language and next they are guided to express it using a simplified notation for OIDs (see figure 6).

Highlight that
"the objects that compose our system collaborate to provide the environment with the required services or to respond to accepted messages"

"we view the world as a set of autonomous agents that collaborate to perform some higher level behaviour." G. Booch

Introduce the concepts
"Community of interacting Objects"
Each object has its own structure and behaviour

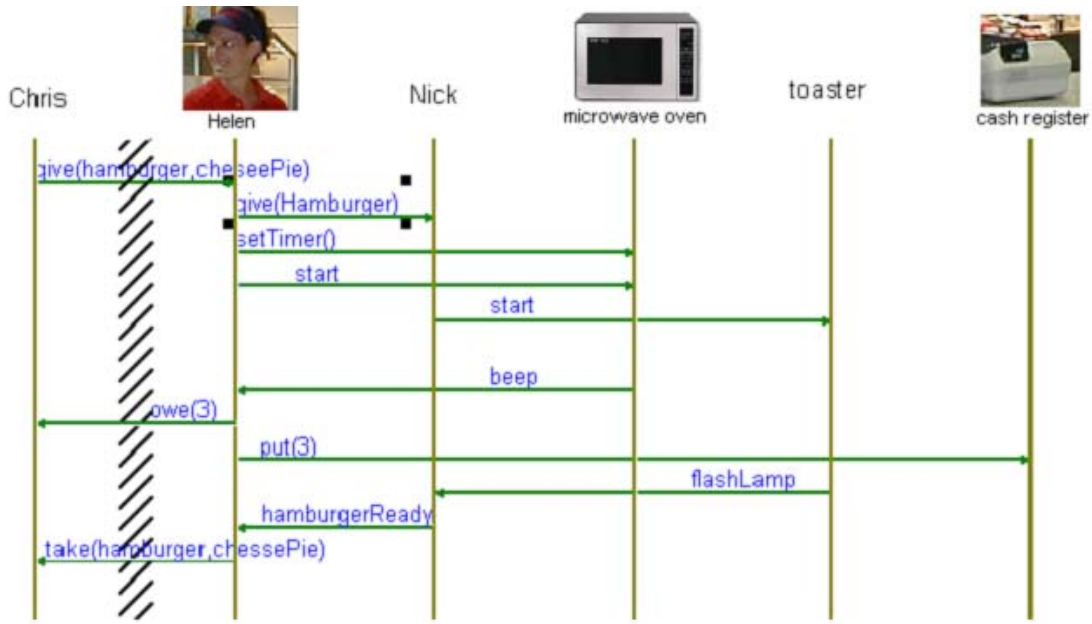

Figure 6. Object Interaction Diagram.

Interface, Implementation, Information Hiding

Consider the following text

"Helen in order to use the microwave oven, only needs to have access to the *timer*, *start,* and *open-door* buttons (see figure 7). We say that these buttons constitute the "interface" of the microwave oven.

Figure 7. Information hiding in microwave oven.

Nick uses only this "interface" and there is no need to know the internals of the microwave oven.

We say that the microwave oven hides its internals, i.e., its implementation, and this is its constructor's decision.

However, its constructor has access to its implementation and of course the same should be true for the technician who is going to repair it.

We have just cited an application of the information hiding principle. All objects, simple or composite, are constructed in such a way that they expose only those items that are involved in the process of getting the messages from the environment or exporting their responses to it. These items must be visible and known to their clients."