

# Object Modeling with UML 2 Finding Classes

February 5, 2013 This is a project to practice on your own what you've learned. ©L J. Waguespack, Ph.D., 2007

## **Telegram Problem Narrative**

#### Description:

A procedure is required to process a stream of telegrams. This stream is available as a sequence of letters, digits and blanks on some device. There exists a buffer of fixed size, into which portions of the input stream can be loaded on demand, and from which characters can be retrieved sequentially. If this buffer becomes empty during execution, it is refilled by transferring the next portion of the input stream into the buffer. Every telegram in the input stream is composed of words that are separated by sequences of blanks, and the telegram is delimited by the reserved word "ZZZZ". The telegram stream is terminated by the occurrence of an empty telegram. An empty telegram is a telegram composed of one or more blanks followed by the delimiter "ZZZZ". The objective of the procedure is to produce a "clean" listing of each telegram accompanied by the chargeable word count and a message indicating the occurrence of an over length word. A "clean" listing of a telegram is an image of the telegram that prints in lines of 120 characters and where the redundant blanks are deleted. All words in the telegram except STOP and ZZZZ are chargeable, and words of more than twelve characters are considered over length.

## **CLASS Finding Exercise**

Given the narrative above you should be able to identify primary issues in the Telegram Problem that would be modeled effectively with classes in a class diagram. Although the problem space may seem to be very "technical" ("computerese!") try to focus on the elements that are of importance to the telegram company, the customers, and the users of this system.

A popular way to look for potential classes is to identify all the nouns (both simple and modified by adjectives) in this problem statement. Among those nouns there will surely be elements of importance to the system that processes telegrams that will need to appear in a class diagram. Make a list and then consider challenging individual nouns as "useful" or "not useful" to understanding the problem.

Once you've pared down the list of potential classes, then go about listing the attributes that each class might need to "remember" in its role in the system. Remember attributes are atomic, so if more than one of anything needs to be "remembered" that will probably require some kind of "collection" based on an association.

Finally, once you have a prospective list of classes, you need to "animate" each class by proposing some behavior that it should provide to the community of objects that will be instantiated by the classes you've identified so far.

This homework doesn't require any diagramming and you can simply use text to record your findings in the project. At the next class you'll be asked to share your findings with other students in the class so be sure to bring your written homework findings.

P.S. Don't spend more than an hour on this homework project !!!

(see What's a buffer? on the next page.)

### What's a "buffer?"

This problem was devised many years ago for students familiar with the details of computer hardware and software. At least one of the terms used is specific to that technology - "buffer."

A buffer is an area of storage (either in hardware or software) that is used to compensate between two different rates of transfer speed. For example you could call a loading dock (where trucks back up to load and unload their cargo) a buffer. It's a "staging area."

A bus station is a buffer because the passengers arrive over a period of time but board the bus generally all together. When they arrive at a destination all the passengers exit the bus together, but may hang around the destinations leaving as they gain other transportation or another bus arrives to continue their journey.

The only thing that might be special about the buffer in this problem is that whatever arrives to go into the buffer maintains that sequence of arrival and leaves the buffer in that exact same sequence (which may not happen with a loading dock or a bus station).