



The Unified Modeling Language (UML)

David M. Rohret

Senior Security Software Engineer, CSC, Inc.

December 6, 2001

Overview

- ◆ What is the Unified Modeling Language
- ◆ Goals of UML
- ◆ UML vs. Object Oriented Models
- ◆ Who Should Use a Modeling Tool
 - Benefits
 - Capabilities
- ◆ Case Studies
- ◆ Available Tools: Free and off the Web





What is the Unified Modeling Language?

- ◆ An Object based modeling tool
 - Seen as a logical follow-on to Object Orientation
- ◆ A marriage of O-O, modeling and development
- ◆ Created as a result of combining the works of Grady Booch and James Rumbaugh, and later Ivar Jacobson



Goals of UML

- ◆ To model systems, not just software
- ◆ To establish an explicit coupling to conceptual ideas as well as executable artifacts
- ◆ To address the issues of extensible inheritance in complex systems
- ◆ To create a modeling language usable by both humans and machines



Goals of UML

(continued)

- ◆ To create usable models
 - Usable models are:
 - Accurate: They correctly describe the the system
 - Consistent: Reduces conflict in views and design
 - Easy to Communicate to Others
 - Easy to Change
 - Understandable: As simple as necessary
- ◆ Standardization of Software and System Modeling



Other OO Methods

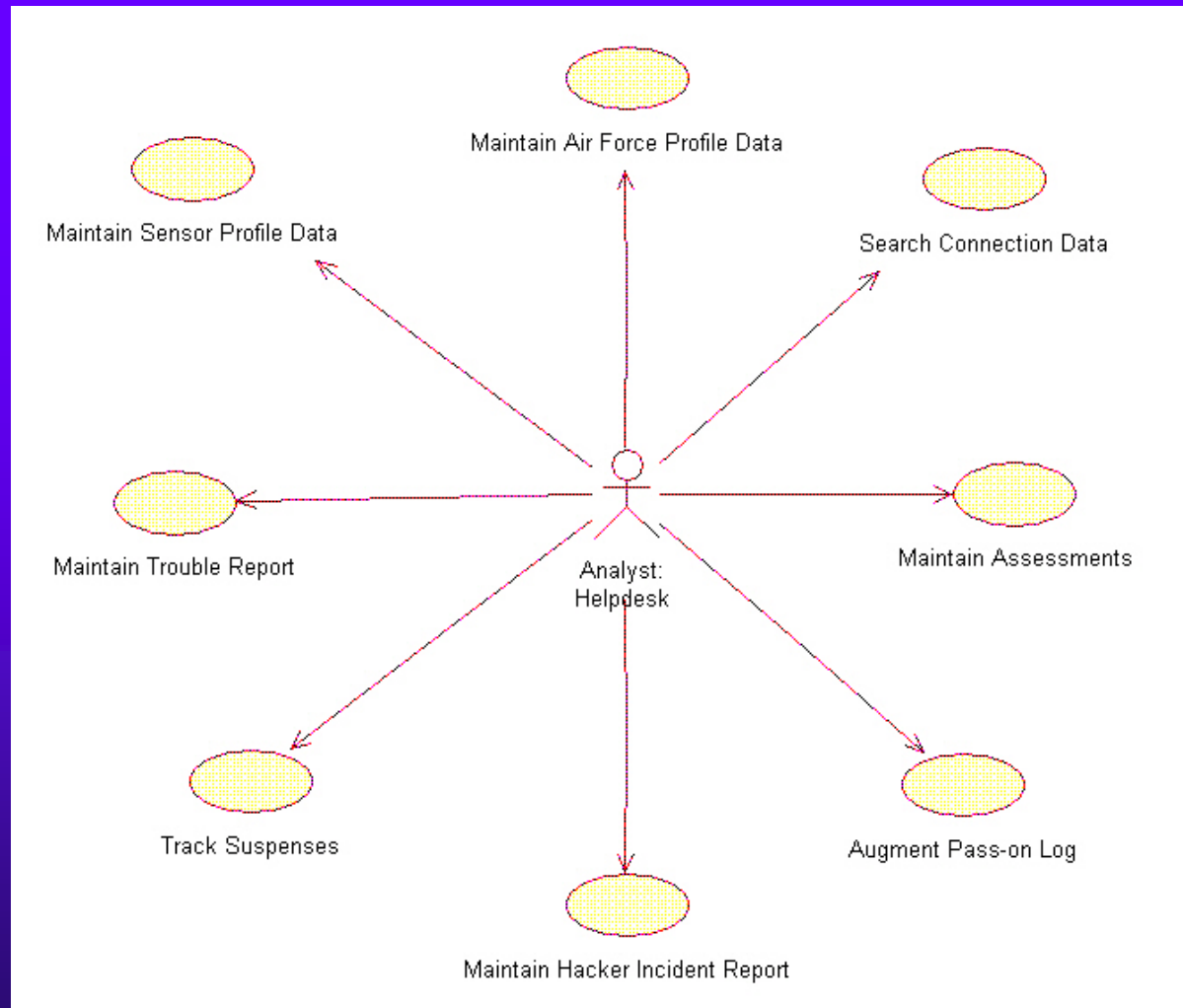
- ◆ Booch: Method of notations
 - Users found symbols difficult to use and understand
 - Highly incremental and iterative process
- ◆ Object Modeling Technique (OMT)
 - Created for software testing based on requirement specifications
 - Object, dynamic, and functional models are encapsulated in Use Cases
- ◆ Objectory Methods
 - Designed for building multiple and diverse systems
 - Hardware, software, and business processes alike

Case Study

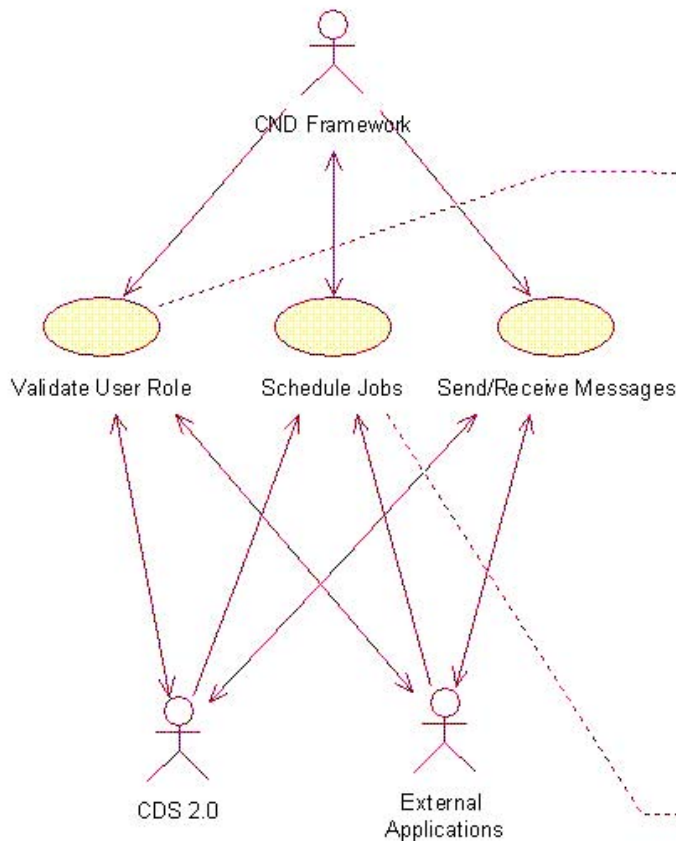
- ◆ An enterprise wide, 4 tier, Java/Oracle system
 - Oracle database
 - Oracle logic
 - Application logic
 - Common interface (Java)
- ◆ Design
- ◆ Testing
- ◆ Deployment
- ◆ Maintenance



Design: Use Case Diagram - Actor



Design: Use Case Diagram – Multiple Actors and Definitions



Validate User Role:

Flow of Events:

1. Application requests permissions key from the Framework, based on user ID.
2. Framework returns permissions key, through which the application's functionality is moderated.

Send/Receive Messages:

Flow of Events:

1. A message producer requests a "topic" to be created, to which the producer will publish messages.
2. Framework returns a topic.
3. A message consumer subscribes to the topic.
4. Producer publishes messages to the topic.
5. All message consumers subscribing to the topic receive the message.

Schedule Jobs:

Flow of Events:

1. Application requests a job be scheduled at a given time and periodicity.
2. Framework schedules the job to be executed as scheduled.
3. Upon completion, the requesting application is sent the completion status.

Who Should Use a Modeling Tool

◆ Benefits

- Forces early requirements and specifications
- Easily understandable for both developers and customers
- Reduces testing time and resources significantly
- Spec. changes and maintenance are less complicated

◆ Capabilities

- Able to model entire system
- Extensible
- Complete view for designers, testers, maintenance and developers
- In-line





Available Tools

◆ Professional Tools

– Rational Rose (integrated system modeling)

- Solaris
- Windows
- \$2K per seat, 10K training for a team of four

– Other

- Solaris
- Windows
- \$6K for site license, 2K each to train

◆ Free (off-the-web)

– Objecteering: http://www.softteam.fr/us/produits_pe.htm

- Windows
- Unix to ship soon