

Fundamentals of Health Workflow Process Analysis and Redesign

Unit 10.3e Process Mapping Unified Modeling Language (UML) notation for Data Flow and Related Diagrams

Component 10/Unit 3e

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 1

Welcome to the Unified Modeling Language notation for Data Flow Diagrams Subunit. This is the fourth Subunit of the Process Mapping Unit.

Topics

- It is not our intent to:
 - cover all of UML
 - cover how to create UML diagrams
- Rather, we introduce three UML diagrams most likely to be encountered by workflow analysts and process re-designers.
- The purpose and notation for each of the three diagrams is covered

There are entire certifications in UML and Object Oriented Design.

It is not our intent to cover this volume of material in this lecture.

Rather, we use this sub-unit to introduce the 3 UML diagrams that are relevant to workflow analysis and process re-design.

We will cover each diagram's purpose and notation.

Upon successful completion of this material the student is able to:

- Describe UML class, activity and state machine diagrams
- Explain the symbols and salient conventions for each of the diagrams
- Interpret the information content and process steps communicated by the diagrams

Remember in the main set of slides for Unit 2 we introduced the concept of ISO 5807 flowcharting notation to diagram a process.

In this subunit, we will provide further details about the ISO 5807 symbols and conventions.

We will describe the standard diagramming symbols and conventions so that you will become familiar with standard flow chart symbols and be able to read a flowchart and understand the data or process flow being communicated.

We will also give examples to help you understand how to use standard symbols to document and communicate a process and then be able to create your own process diagrams using standard flowchart form.

Methods for Diagramming Processes

| Process Aspects | ISO 5807 | Yourdon | Gane-Sarson | UML | E-R Crows-foot |
|------------------------|----------|-----------|-------------|-----|----------------|
| Context | | X | X | X | |
| Process steps | X | | | X | |
| Data flow steps | X | X | X | X | |
| Information content | | text only | text only | X | X |
| Data transformation | X | X | X | X | |
| Flow control and state | X | | text only | X | |
| Roles involved | X | | | X | |

Each of the methods for diagramming a process covers different process aspects.

Unified modeling language (UML) has diagrams to represent each of the process aspects that we are interested in for process analysis and re-design. These include use case diagrams that represent the context in which the system operates, process and data flow steps, information content, any information transformations that occur or should occur, the order or sequence of the steps involved in the process including flow control and state, and the roles of the persons completing the steps in the process. There are 13 different types of UML diagrams, we will cover three of them here.

Background

- Developed by the Object Management Group (OMG)
- First version, UML1 released in 1997¹
- Created to overcome incompatibility of different notations for software modeling
 - i.e., for software design and development, NOT workflow or process representation.
- Adapts and incorporates pre-existing notations:
 - flowcharts, Yourdon, Gane-Sarson, Entity-relationship diagrams and ISO 5807.

Unified Modeling Language is a standard developed and maintained by the Object Management Group (OMG)

The first UML standard, UML1 was released in 1997¹

OMG states the rationale behind the creation of the UML standard was that, “visual software modeling was plagued by the incompatibility of different notations created by different modelling gurus. The absence of a standardized notation deterred potential users, and as an inevitable result the modeling tools market was tiny and fragmented”. UML is for software design and development, NOT workflow or process representation. UML adapts and incorporates notations that existed in the early 1990’s, e.g. ISO 5807 Flowcharts, Yourdon, Gane-Sarson, entity-relationship diagrams.

Use

- Developed for visual modeling in software development
- Later, to automate software development
- 13 different diagrams for representing aspects of computer software, e.g., functionality, data content, and data movement
 - NOT physical process steps
- Many IT professionals are trained in UML
- Workflow analysts may encounter UML diagrams

UML notation was developed to support visual modeling in the development of software, and later, to support using the models to automate software development.

UML provides different diagram types that represent aspects of computer software, e.g., functionality, data content, and data movement, NOT physical process steps.

Many IT professionals are trained in UML, thus workflow analysts may encounter UML diagrams in practice. Thus, we present in this module how to READ UML diagrams rather than detailed training on how to create UML models.

Types of UML 2.0 Diagrams

Structure Diagrams (six types) represent static aspects of system structure. We will cover the Class Diagram.

Behavior Diagrams (three types) represent dynamic aspects of systems. We will cover two of the three, the Activity Diagram, and State Machine Diagram.

Interaction Diagrams (four types) are detailed behavior diagrams. We will not cover interaction diagrams.

There are different types of UML diagrams.

We will cover 1 type of **Structure Diagram** -- the Class Diagram.

We will cover 2 types of **Behavior Diagrams** -- the Activity Diagram, and State Machine Diagram. and

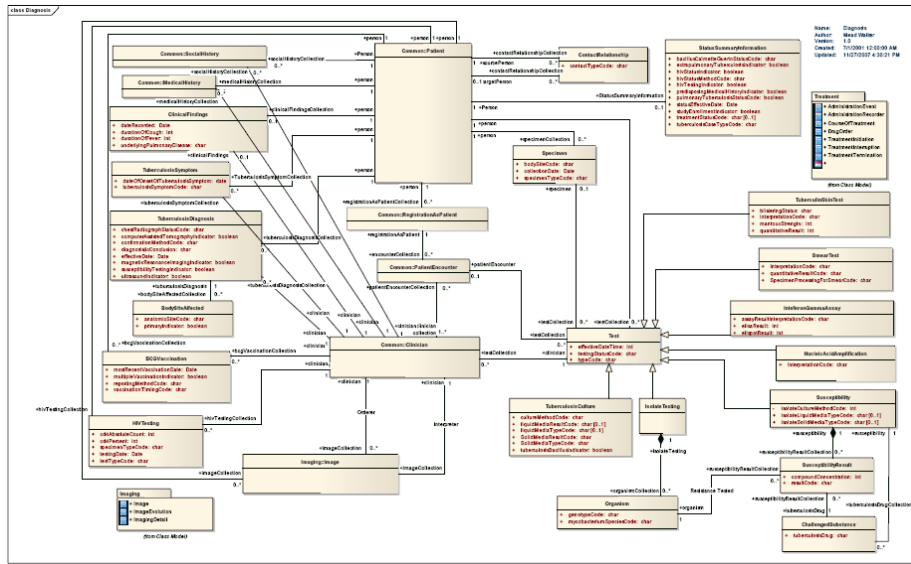
We will not cover **Interaction Diagrams** in this lecture.

Class Diagram

- A class diagram is a model of the information stored in a system
- A class diagram is analogous to the E-R diagram

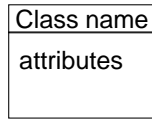
A class diagram is a model of the information stored in a system and is analogous to the E-R diagram

Class Diagram Example

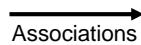


A class diagram shows static information content, for example, data that are collected or used in patient care. Data collected and used in the diagnosis of tuberculosis are shown on the diagram in the slide. Class diagrams show the data in boxes and the relationships between the data, called associations. Like entity-relationship diagrams, class diagrams also show the minimum and maximum times an instance of one class can be related to an instance of another class, called cardinality and modality of the relationships.

Class Diagram Notation



Classes are shown by boxes



Associations are shown by lines

0..1
1..*

Cardinality and modality are shown by numbers on the lines

A class, shown by a box, represents the information collected, generated by or used in a process.

An association, shown by a line, represents relationships between the information. Where an entity-relationship diagram shows only the number of times data can be related, class diagrams employ more types of relationships, including aggregation, composition and inheritance that are designated by different arrow heads. Navigability, i.e. direction of the relationship is shown by an arrow head.

Cardinality and modality of the association are shown by numbers on the lines.

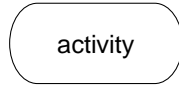
Activity Diagram

- Similar to a flowchart but fewer symbols
- Shows movement
- Uses fewer symbols than flow charts

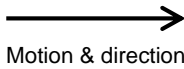
Similar to a flowchart, an activity diagram shows movement of either tasks/steps or movement of information. In UML parlance, this is called the dynamic aspect of a process, or simply “behavior” as opposed to class diagrams that show the “static” aspects.

Although activity diagrams have a significant number of features in common with flow charts, the symbol set is smaller. Activity diagrams focus on the movement rather than using symbols to differentiate types of data, or types of task steps, etc.

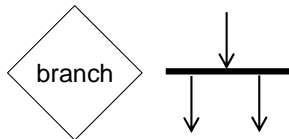
Activity Diagram Symbols



Activities are shown by a long rectangle with rounded corners



Motion and direction are shown by lines with arrow heads



Branch points are shown by *either* diamonds or heavy horizontal lines



Start of a process is shown by a solid circle



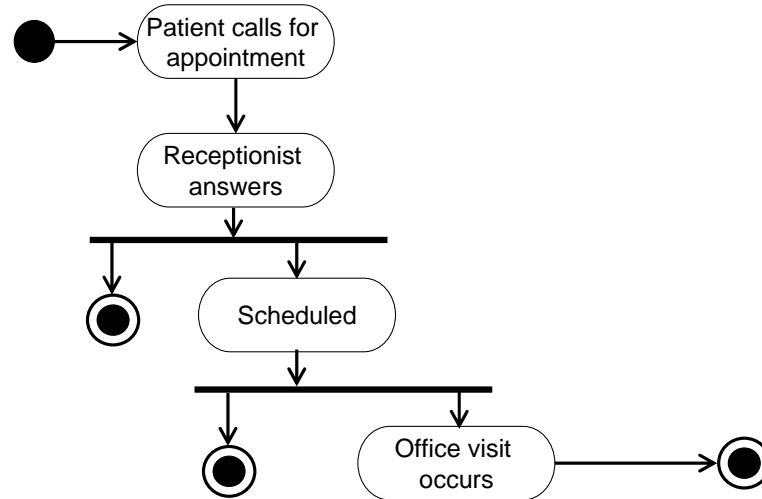
End of a process is shown by a solid circle concentric with a "no fill" circle

An **activity**, shown by a long rectangle with rounded corners, is a task that is performed either by a person or a computer system.

Motion and direction, shown by lines with arrow heads, indicate the flow of the process or information.

Branch points in a process, shown by either a diamond or a heavy horizontal line, indicate when more than one option or path exists by which the process can proceed or the information can flow. The start and stop of a process or information flow, are shown by circles.

Activity Diagram Example



Component 10/Unit 3e

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 14

In this simplified activity diagram example, the process starts when a patient calls to schedule an office visit. The receptionist answers and attempts to schedule the appointment. If a mutually convenient time is found, the appointment is scheduled, if not, the process ends. When the scheduled date arrives, the patient attends the visit (or not). The process is complete when the office visit is over.

State Diagram

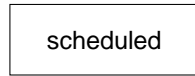
- Shows the different “statuses” allowed
- Valid transitions from state to state

A **state** machine diagram, also called state diagram, shows the different states, also called “statuses”, allowed as a thing moves through a process, or as information is processed by a system.

For example, a prescription may have any of the following statuses:

- Requested by patient
- Authorized by provider
- Sent to pharmacy
- Filled
- Not filled

State Diagram Symbols



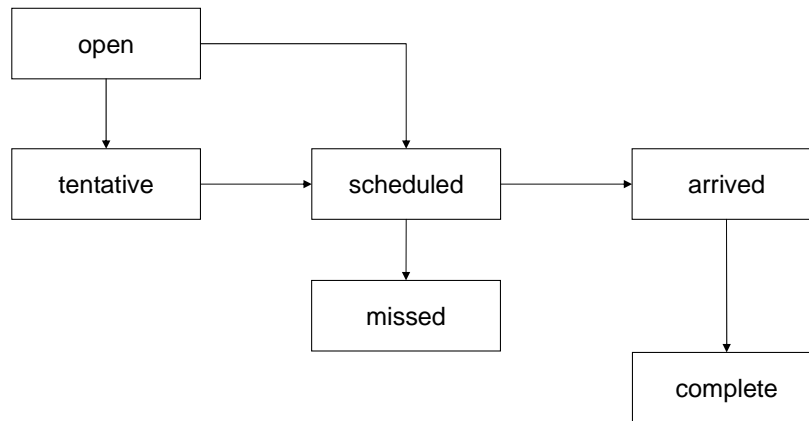
Rectangles that show the states



Lines and arrows that show valid transitions from state to state

State diagrams symbols are simply rectangles and lines and arrows that denote transition.

State Diagram Example



Component 10/Unit 3e

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 17

This simplified state diagram example corresponds to the previous activity diagram example. The process starts when a patient calls to schedule an office visit. The receptionist answers and attempts to schedule the appointment. While trying to find a mutually convenient time, candidate time slots are marked as tentative. When a time slot that will work for both the patient and the practice is found, the appointment is scheduled. That state remains until the appointment is arrived (the patient has arrived). At the end of the time slot, visits that have not been arrived are marked as missed. If the time slot has been marked as arrived, it is given a status of complete when the claim form is entered, i.e., when the patient is leaving.

Importantly, the diagram shows valid transitions, for example, according to the diagram example, a scheduled slot can go to states of arrived or missed, but a scheduled time slot can not go directly to a status of complete.

Maintenance

- UML is an international standard and is maintained by the OMG.
- The standard is freely available from the OMG web site

UML is an international standard and is maintained by the OMG.
The standard is freely available from the OMG web site

References

- 1.) **Watson, Andrew Visual Modelling: past, present and future. Accessed June 22, 2010. available from http://www.uml.org/Visual_Modeling.pdf**

Suggested Reading

UML overview on OMG homepage:

<http://www.uml.org/>

UML Wikipedia topic:

http://en.wikipedia.org/wiki/Unified_Modeling_Language

These references are provided for your use.