SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

- ❑ **What is UML ?**

  A general –purpose visual modeling language.

- ❑ **UML Usage**

  To specify , visualize , construct and document the artifacts of a software system.

- ❑ **During a new system construction**

  It captures decisions and understanding in the requirement phase.

- ❑ **Usage in terms of software development process**

  Use to understand, design, browse, configure, maintain and control information about a system.

IV. Whence and whither UML? Big picture and history

- **UML usage intension in a bigger picture**
  - **For usage with development methods, development lifecycle stages, application domain, and media.**
  - **To unify past experience about modeling techniques and to incorporate current software best practices into a standard approach.**
- **UML include**

  Semantic concepts, notation, guild lines.
- **UML parts**

  Static, dynamic , environmental and organizational parts
- **Supporting tools**

  Interactive modeling tools that has code generator and report writer.

IV. Whence and whither UML? Big picture and history

- **UML Specification**
  Does not define a standard process but useful with iterative development process.
  Usefulness = UML + interactive development process
  Supporting most OOD processes
- **UML information capturing (groupings)**
  Static structure and dynamic behavior of a system(as 2 UML parts).
- A system is modeled as a collection of discrete objects that interact to perform work.
- **Static Structure**
  Defines the type of object important to the system.
- **Dynamic behavior :** Defines the history of objects over time and the communication among objects to accomplish goals.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

- **Modeling a system from viewpoints (separately but related)**

  Permits the understanding of the system for different purposes.

- **Organizational constructs (one of the UML part)**

  It permits arranging models into Packages – the partitioning of large system into workable pieces.

  - **Best management**

    The control of dependencies among packages, management of version of model units.

  - **UML different from Programming language**

    UML lacks syntactic and semantic organization that most programming languages have.

- **UML Tool capability**
- Code generation : UML to a variety of programming languages.
- Reverse engineering : Existing code to Model.

IV. Whence and whither UML? Big picture and history

- **UML History**

Efforts that led to UML development:

To simplify and consolidate large number of Object Orientation method that had emerged.

- **Object –oriented development methods**

- Development method for traditional programming languages(Cobol and Fortran) Emerges in the 1970s and widespread in the 1080s.
  - Development Method

    Structured analysis and Structured design and its variants, such as Real Time Structured Design and others developed by Constantine, DeMarco, Mellor, Ward,…

- **Achievements :**Penetration in large system area, especially of government contracted systems in the aerospace and defense fields.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

- ❑ **UML History**

Development methods for traditional programming languages

In the large system area, contracting officers insisted on an organized development process and ample documentation of the system design and implementation.

- Results were not always as good as hoped for

Many computer-aided software engineering (CASE) system were little more than report generators that extracted design after the implementation was complete.

- Commercial applications were more reluctant to adopt large CASE systems and development methods.

- Most businesses developed software internally for their own needs.

Commercial systems were perceived to be simpler and there was less need for review by outside organizations.

IV. Whence and whither UML? Big picture and history

❑ **UML History**

Development methods for Object oriented Languages

❖ **Simula, developed in Norway in 1967**
- Acknowledged as the first Object oriented language.
- It greatly influenced the developers of several Object oriented languages but did not have a significant following.
- it was the work of Dahl and Nygaard.

❖ **Smalltalk in the early 1980s**
- Made the Object Oriented movement spread widely.

❖ **Other Object oriented Languages such as Objective C, C++, Eiffel and CLOS.**

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ **UML History**

Development methods for Object oriented Languages

❖ **First Object orientated development method**

- Published by Shlaer/Mellor and Coad/Yourdon

❖ **Second Object oriented development method**

-Published by Booch,Rumbaugh/Blaha/Premerlani/Eddy/Lorensen and Wirfs-Brock/Wilkerson/Wiener

- There were several papers publication and books there after.

❑ **Unification effort**

- There were early attempts to unify concepts among methods.

Example : **Coleman and his Colleagues with concepts from OMT, Booch and CRC. It was regarded as another method as it did involve the original authors.**

IV. Whence and whither UML? Big picture and history

❑ **UML History**

Development methods for Object oriented Languages

❖ **Successful attempt to combine and replace existing approaches**
- When Rumbaugh joined Booch at Rational Software Cooperation in 1994.
- They began combining the concepts from OMT and Booch Methods, resulting in a first proposal in 1995.
- Jacobson also joined Rational and began working with Booch and Rumbaugh.
- Their Joined work was called Unified Modeling Language(UML).

❖ **Request for Proposal for standard approach to Object oriented modeling .**
- Send by OMG (Object Management Group) in 1996

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ **UML History**

Development methods for Object oriented Languages

❖ **UML authors reaction to the request proposal**

- **Booch, Jacobson and Rumbaugh began working with methodologist and developers from other companies to produce a proposal.**
- **Proposal with properties**
  - ➢ Attractive to the membership of OMG
  - ➢ Modeling language that would be widely accepted by tool makers, methodologist and developers who will use it.
- The proposal was submitted to the OMG in September 1997.
- The final product is a collaboration among many people.

IV. Whence and whither UML? Big picture and history

- **UML Standardization**

  Development methods for Object oriented Languages

- **UML was accepted by the Membership of the OMG as a standard in November 1997.**
- **The OMG assumed responsibility for the further development of the UML standard.**
- **Other works**
  - **A number of books were published outlining the highlights of UML.**
  - **Many vendors announced support for the UML**
  - **Several methodologists announced they will use UML notation.**
- **Reasons for its attraction**
  - **It consolidates the experiences of many authors with an official status .**

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ **UML Standardization**

Development methods for Object oriented Languages

❖ **A series of internal research conference with the title UML yyyy.**

-Where yyyy is a year starting with 1998 and continually manually**.**

❑ **UML 2**

❖ **OMG issued request for proposal to upgrade UML**

- After several years of experiences using UML
- To fix problems uncovered by experience of use
- To extend it with additional capabilities that were desired in several application domains.

❖ **November 2000 to July 2003**

- Proposals were developed
- A specification of UML version 2.0 we being adopted by OMG membership.

IV. Whence and whither UML? Big picture and history

❑ **UML Standardization**

Development methods for Object oriented Languages

❑ **UML 2**

❖ **OMG finalization process**

- To fix bugs and problems encountered in initial implementation.
- Final specification expected at the end of 2004 or beginning 2005

❖ We consider UML2 as UML version 2.0 and UML1 as first version.

❖ New features

- UML2 is like UML1 in most commonly used, central features.
- In UML2, a few major enhancements have been added, and many small bugs have been fixed, but user of UML1 has no problems in using UML2.

IV. Whence and whither UML? Big picture and history

- ❑ **UML Standardization**

  Development methods for Object oriented Languages

- ❑ **UML 2**
- ❖ New features
  - ▪ Some of the Most important changes visible to user are:-

    1. Sequence diagram constructs and notation based largely on the ITU message sequence chart standard, adapted to make it more object oriented.
    2. Decoupling of activity modeling concepts from state machine and the use of notation popular in the business modeling community.
    3. Unification of activity modeling with the action modeling added in UML version 1.5 to provide a more complete procedural model.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

- **UML Standardization**

  Development methods for Object oriented Languages
- **UML 2**

- New features
  - Some of the Most important changes visible to user are:-

  4. Contextual modeling constructs for the internal composition of classes and collaborations.
  5. Repositioning of components as design constructs and artifacts as physical entities that are deployed.
- Internal mechanisms

  - Changes that affect the internal representation of UML constructs and its relationship to other specifications.

IV. Whence and whither UML? Big picture and history

- ❑ **UML Standardization**

  Development methods for Object oriented Languages

- ❑ **UML 2**

- ❖ Internal mechanisms

  - Changes that affect the internal representation of UML constructs and its relationship to other specifications.
  - Changes may not concern most users directly but are important to toolmakers because they affect interoperability across multiple specification. Therefore it will affect users indirectly.

1. Unification of the core of UML with the conceptual modeling parts of MOF (Meta-Objet Facility).
2. Restructuring of the UML meta model to eliminate redundant constructs and to permit reuse of well defined subsets by other specification.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

- ❑ **UML Standardization**

  Development methods for Object oriented Languages

- ❑ **Other sources**

  - Certain UML views show strong influences form particular non object oriented sources

1. The static view,
2. State machine models
3. The sequence diagram notation
4. The structured classifier
5. The activity diagram notation of UML1

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ What does unified mean?

❖ **The following relevant meanings of UML**

1. **Across historical methods and notations**
2. **Across the development lifecycle**
3. **Across application domains**
4. **Across implementation languages and platforms**
5. **Across development processes**
6. **Across internal concepts**

IV. Whence and whither UML? Big picture and history

❑ Goals of UML

- General purpose modeling language that all modelers can use.
- Nonproprietary and based on common agreement by much of the computing community.
- Intended to supersede the models of OMT, Booch, and Objectory , as well as those of other participants of the proposal.
- Intended to be familiar as possible, whenever possible using several notation from the contributors.
- Meant to support good practices for design,
  - Encapsulation, separation of concern,
- And capture the intend of a model construct
- Intended to address current software development issues. Such as large scale, distribution, concurrency, patterns, and team development.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ **Goals of UML**

❖ **What UML was not intended**

- To be a complete development method
    - it does not include step by step development process.

❖ **What UML is intended**

- To support all, or at least most, of the existing development process to build strong architecture to solve user case driven requirements.
- To be simple as possible while still capable of modeling a full range of practical systems.
- Be able to handle all the concepts that arise
    - In a modern system: **Example**: concurrency and distribution
    - In software engineering mechanisms

      Example : encapsulation and components

It has several kinds of models, but you don’t have to learn it all at once.

IV. Whence and whither UML? Big picture and history

❑ Complexity of UML

- It a large and varied modeling language intended for use on many different levels and at many stages of the development lifecycle.
- It has be criticized for being large and complex but the complexity is for it general purpose application.
- The complexity must be understood in the light of its history.

1. UML is a product of consensus of persons with varied goals and interests.
2. It was originally the merger of leading modeling approaches and later has been the target for accommodating a number of existing notations.
3. The official specification document have been written by teams of uneven ability.

IV. Whence and whither UML? Big picture and history

❑ Complexity of UML

4. UML is not a precise specification in the manner of a formal language.
5. The semantics sections sometimes contains vague statements without adequate explanation and examples
6. There is far too much use of generalization at the expense of essential distinctions.
7. There is a tension between concepts for conceptual modeling and programming language representation, with no consistent guidelines.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ UML Assessment

- ❖ UML is messy, imprecise, complex, and sprawling. That is both a fall and a virtue.
- ❖ You don't have to know or use every feature of UML any more than you need to know or use very feature of a large software application or programming language.
- ❖ UML is more than a visual notation. It models can be use to generate code and test cases.
- ❖ It is unnecessary to listen too much to UML language lawyers. There is no single right way to use it.

- It is one of many tools that a good developers uses.
- It doesn't have to be used for everything.
- You can modify it to suit your own needs provided you have the cooperation of your application of your colleagues and software tools.

IV. Whence and whither UML? Big picture and history

❑ **UML Concept Areas**

- **UML concepts and models can be grouped into the following concept areas.**

1. Static structure

-Defines the universe of discourse (grouped as static view)

- The key concepts form the application
- Their internal properties
- Their relationship to each other

- **Application concepts are modeled in classes**

- Each describes discrete objects that hold information and communicate to implement behavior.
- Information they hold modeled as attribute
- Behavior perform modeled as operations
- Several classes share common structure using generalization.

IV. Whence and whither UML? Big picture and history

❑ **UML Concept Areas**

- **UML concepts and models can be grouped into the following concept areas.**

1. Static structure

- **Application concepts are modeled in classes**
  - A child add incremental structure and behavior to the structure and behavior obtained by inheritance from the common parent class.
  - Object have runtime connections to other individual objects.
  - Object to object relationships are modeled as association among classes.
  - Dependency relationship - relationship among elements
  - Classes may have interfaces, which describe their externally visible behavior.

IV. Whence and whither UML? Big picture and history

❑ **UML Concept Areas**

- **UML concepts and models can be grouped into the following concept areas.**

1. Static structure

- **Application concepts are modeled in classes**

• Static view is noted for using class diagram and its variants.

• Use to generate most data structure declarations in a program.

• Other elements in UML diagrams – interfaces, data types, use cases, and signals collectively called classifiers.

IV. Whence and whither UML? Big picture and history

- **UML Concept Areas**

  - **UML concepts and models can be grouped into the following concept areas.**

  2. Design constructs

- UML models are meant for both **Logical Analysis and design intended for implementation.**

IV. Whence and whither UML? Big picture and history

❑ UML Concept Areas

- **UML concepts and models can be grouped into the following concept areas.**

3. Deployment constructs

- UML models are meant for both **Logical Analysis and design intended for implementation.**

IV. Whence and whither UML? Big picture and history

❑ **UML Concept Areas**

- **UML concepts and models can be grouped into the following concept areas.**

4. Dynamic behavior

❖ There are 3 ways to model behavior

1. Life history of one object as it interacts with the rest of the world
2. The communication pattern of a set of connected objects as they interact to implement behavior
3. The evolution of execution process as it passes through various activities.

❖ View of an object in isolation is a state machine

- A view of an object as it respond to event based on its current state.
- Performs actions as part of its response, and transition to new state.

State machine are displayed on state machine diagrams.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ UML Concept Areas

- **UML concepts and models can be grouped into the following concept areas.**

4. Dynamic behavior

Interaction

- An interaction overlays a collaboration with the flow of message between parts.
- Interaction are shown in the sequence diagrams and communication diagrams.
- Sequence diagram emphasis time sequence whereas communication diagrams emphasis object relationships.

Activity

- it represents the execution of a computation.
- Modeled as a set of note connected by control flows and data flows.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ **UML Concept Areas**

- **UML concepts and models can be grouped into the following concept areas.**

4. Dynamic behavior

Activity

- Can model both sequential and concurrent behavior.
- it include decisions and loops.
- Activity diagram may be use to show computations as well as workflows in human organizations.

Use case

- Guiding all the behavior view
- Each case is a description of a slice of system functionality as visible to an actor, an external user of the system.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ UML Concept Areas

- **UML concepts and models can be grouped into the following concept areas.**

4. Dynamic behavior

Use case

- Use case view includes both
  - Static structure of the use cases and their actors
  - Dynamic sequence of messages among actors and system, usually expressed as sequence diagrams or just text.

IV. Whence and whither UML? Big picture and history

- ❑ **UML Concept Areas**

- UML concepts and models can be grouped into the following concept areas.

5. Model organization

- Modeling information must be divided into coherent pieces so that teams can work on different parts concurrently.

- Humans needs the organization of model content into packages of modest size.

Packages

- These are general purpose hierarchical organizational units of UML models.
- Can be used for storage, access control, configuration management, and constructing libraries that contain reusable model fragments.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ **UML Concept Areas**

- **UML concepts and models can be grouped into the following concept areas.**

5. Model organization

Packages

- Dependency between packages summaries dependences among package content.
- Dependency among package can be imposed by the overall system architecture.
- The content must conform to the package dependencies and to the imposed system architecture.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

❑ **UML Concept Areas**

- **UML concepts and models can be grouped into the following concept areas.**

6. Profiles

- No matter a language completeness people will want to make extensions.
- UML contains limited extensibility capability to accommodate day to day needs for extensions with requiring change to basic language.

Stereotype

- A new kind of model element with the same structure as an existing element.
- With additional constraints , a different interpretation and icon
- Different treatment by code generators and other back end tools.
- It defines a set of tagged values.

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

- ❑ **UML Concept Areas**

  - **UML concepts and models can be grouped into the following concept areas.**

  6. Profiles

Stereotype

- It defines a set of tagged values.
- A tagged value is a user defined attribute that applies to model elements themselves rather than objects in run time system.

Example :

Tagged values may indicate project manager information, code generator guidance, and domain specific information

SE 3140: SOFTWARE DESIGN AND MODELING

IV. Whence and whither UML? Big picture and history

- **UML Concept Areas**

  - **UML concepts and models can be grouped into the following concept areas.**

6. Profiles

Constraint

- well-formedness condition expressed as a text string in some constraint language- Programming language, special constraint language or natural language.
- UML includes a constraint language called OCL.

❖ A profile is a set of stereotypes and constraints for a particular purpose that can be applied to user package.

❖ Profiles can be developed for particular purposes and stored in libraries for use in user models.

❖ It must be use with care so that others may understand.