

Object Oriented Modeling

Overview of the Object Management Group (OMG)

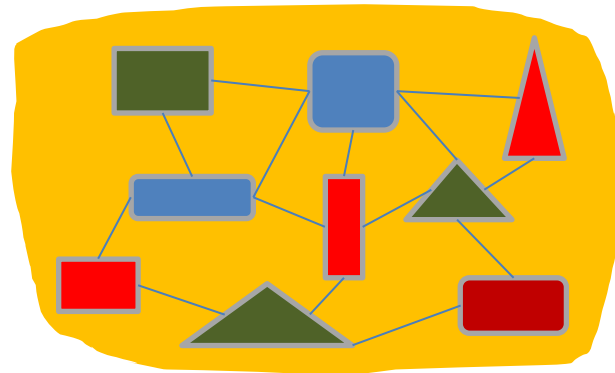
“A model is the explicit interpretation of one’s understanding of a situation, or merely of one’s ideas about that situation. It can be expressed in mathematics, symbols or words, but it is essentially a description of entities, processes or attributes and the relationships between them. It may be prescriptive or illustrative, **but above all, it must be useful.**”

Brian Wilson. Systems: Concepts, Methodologies and Applications. John Wiley and Sons, 1990, 2nd Edition, page 11

Structure and Behaviour

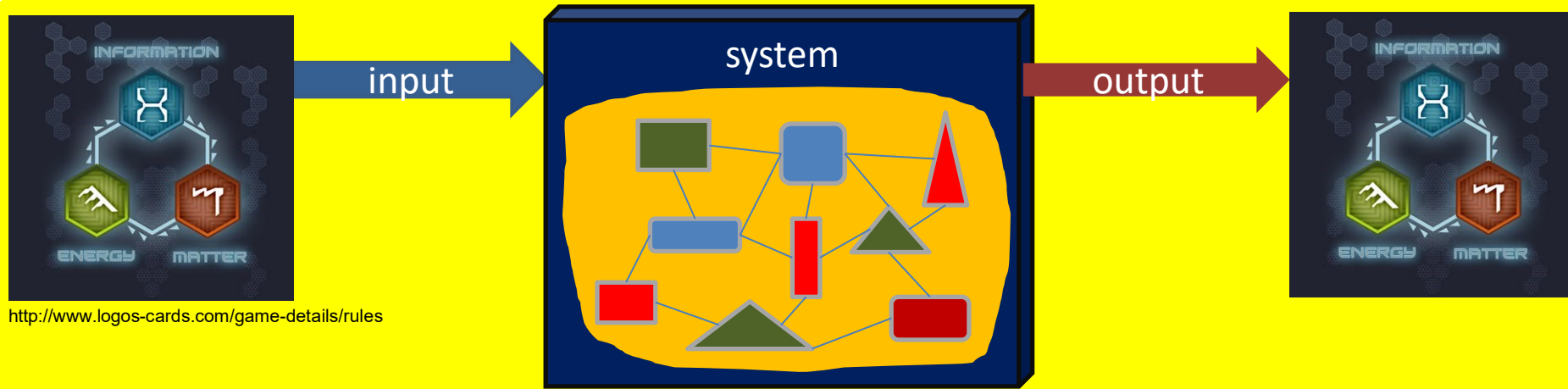
“The structural schema defines the types of the facts contained in the information base. These facts change over time, but they cannot change in any arbitrary way. Only some changes of the information base are permissible. These changes are called domain events. **The definition of the domain event types is the most important part of the behavioral schema.**”

Conceptual Modeling of Information Systems, page 247



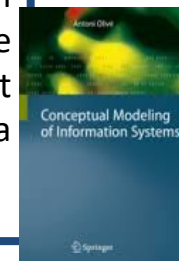
Systems and States

- **Memorize**: to maintain a representation of the state of its **domain**.
- **Inform**: to provide information about the state of its **domain**.
- **Act**: to perform actions that change the state of its **domain**.

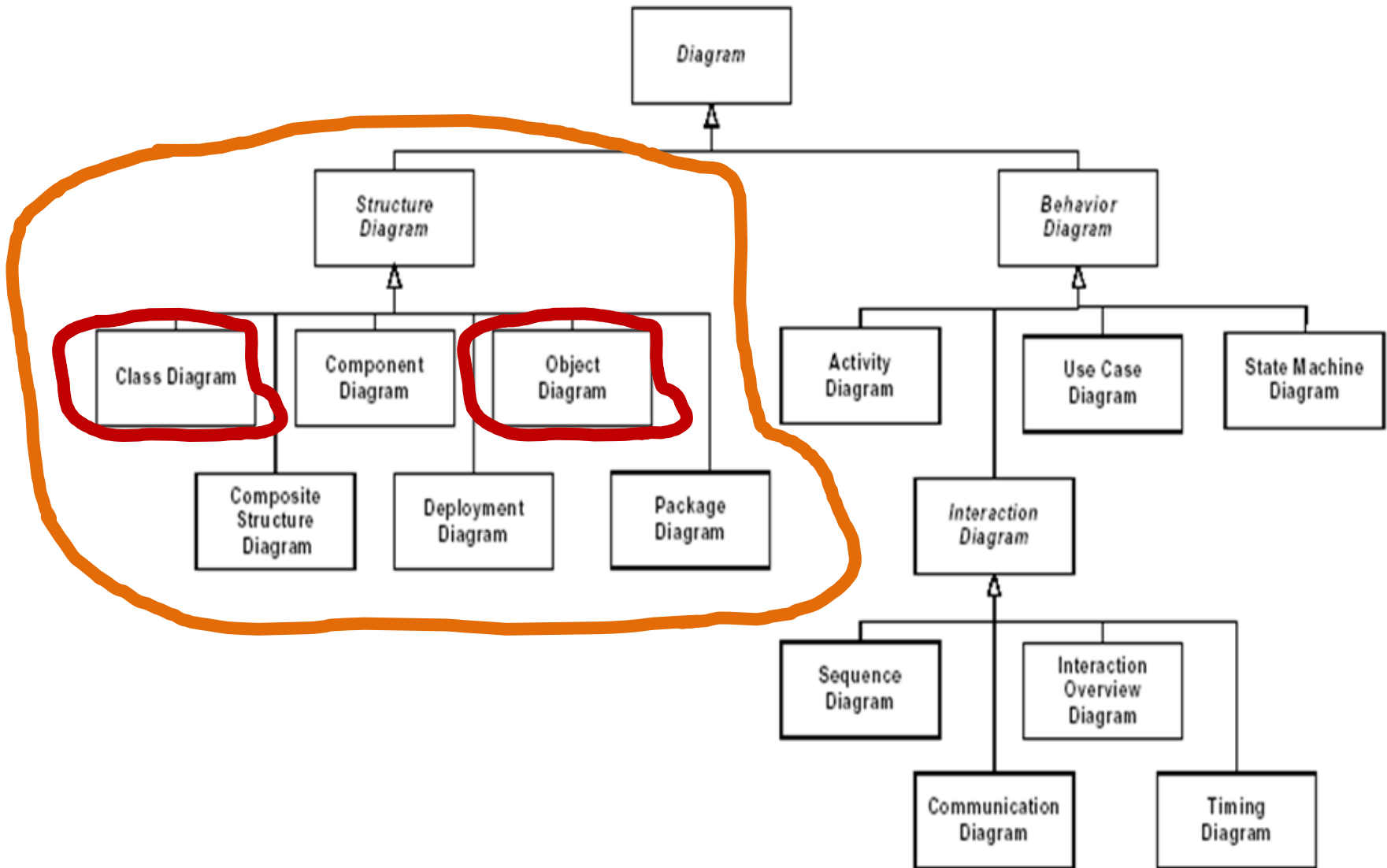


(...) The **state of a domain** at any given point in time is the set of instances of the relevant entity and relationship types that exist in the domain at that time. (...) Most domains change over time, and therefore their state changes too. When the state of a domain changes, the information base must change accordingly. Of course, a domain cannot change in an arbitrary way. Only some changes are acceptable. The acceptable changes are called **domain events**. This concept (of a domain event) can be defined precisely in terms of a more basic concept, called a structural event (...)

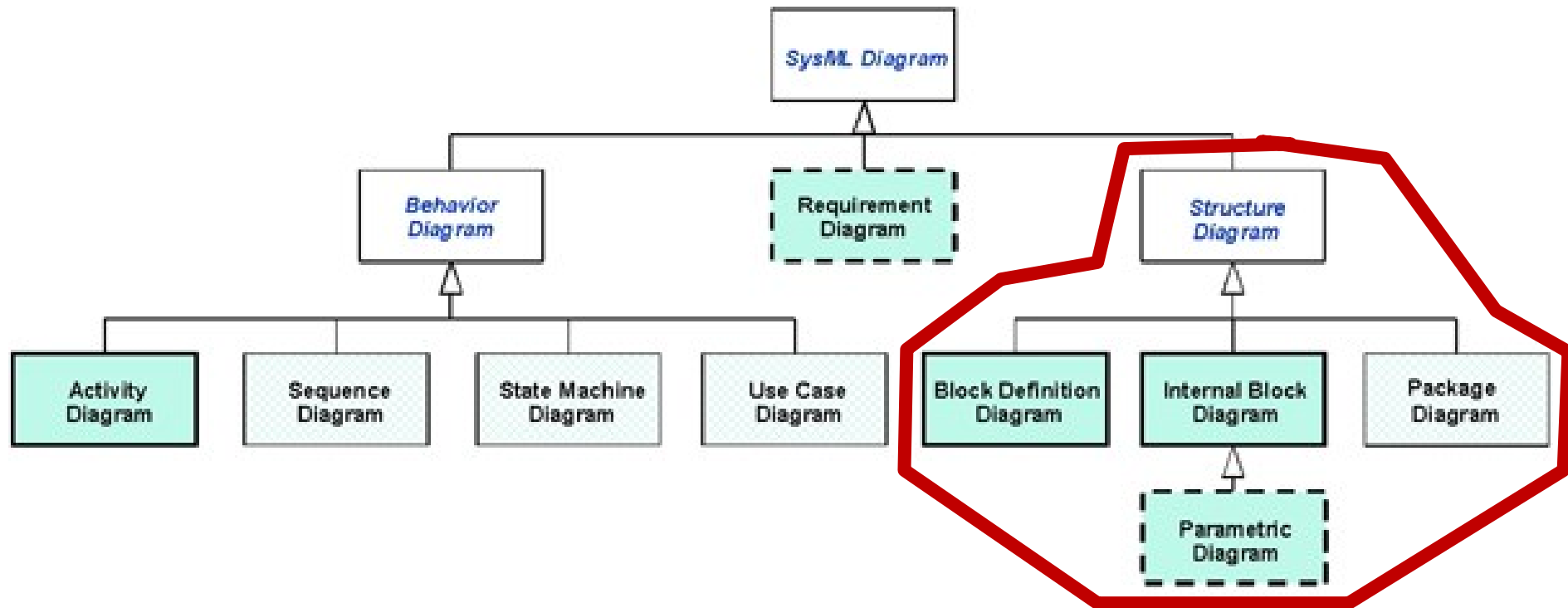
Conceptual Modeling of Information Systems, page 248



Structure Modelling in UML

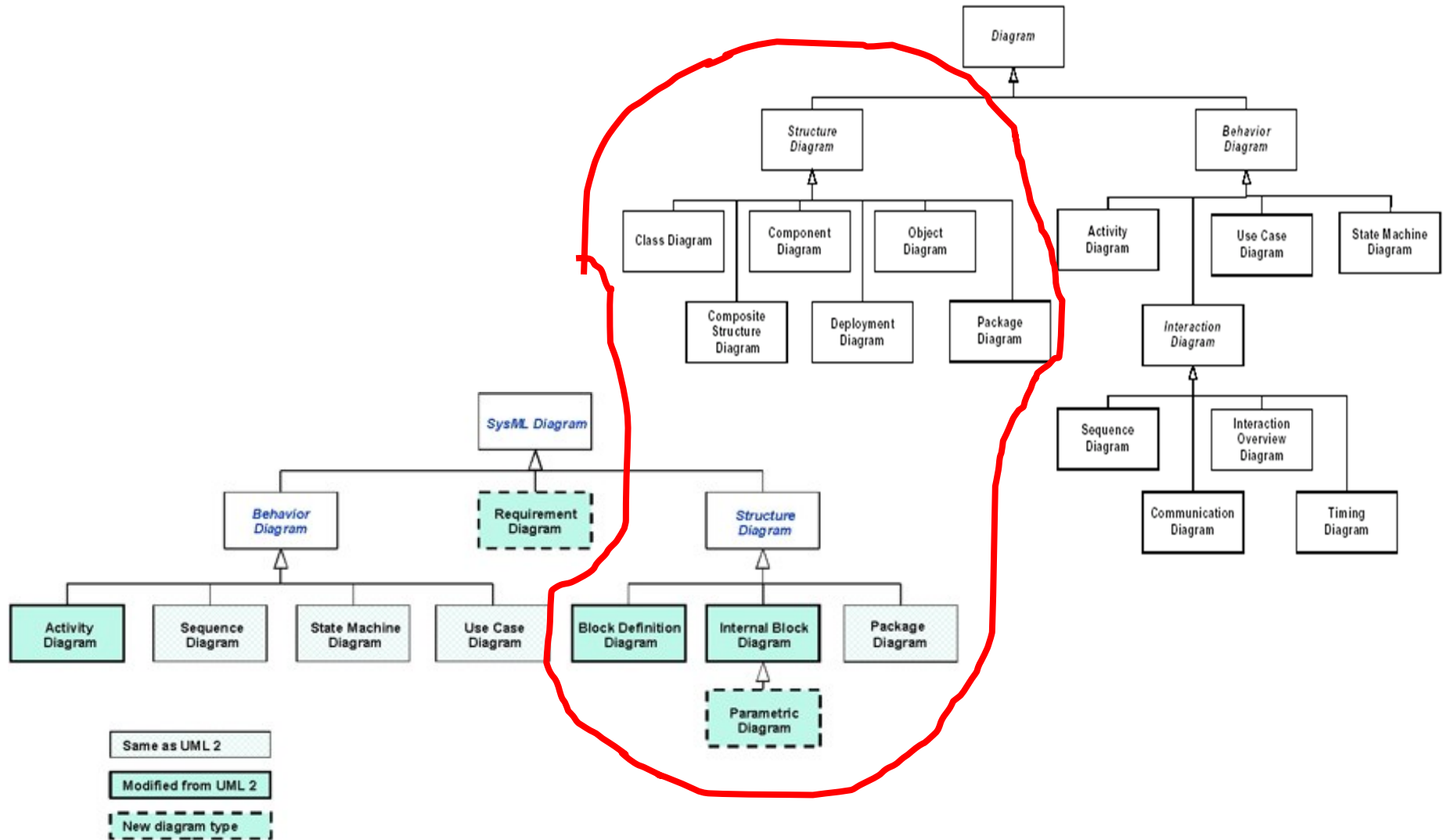


Structural Diagrams in SysML



- The «**block**» is the **basic unit of structure** in SysML.
- It can be used to represent any kind of hardware (including facilities, persons, etc.), software, or any other system element.
- The system structure is represented by **block definition diagrams** and **internal block diagrams**.

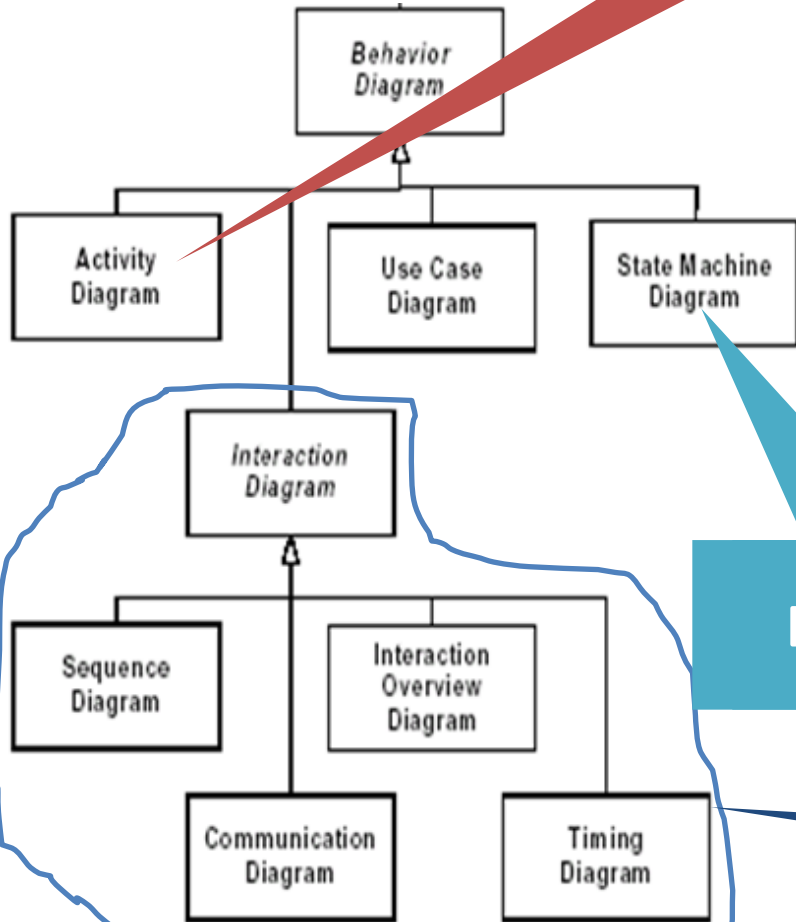
Structural diagrams in UML and SysML



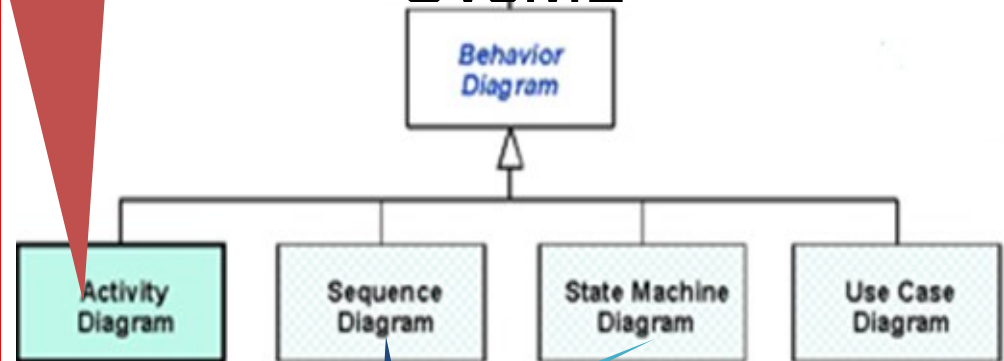
Behavior in UML and SysML

Flow-based

UML



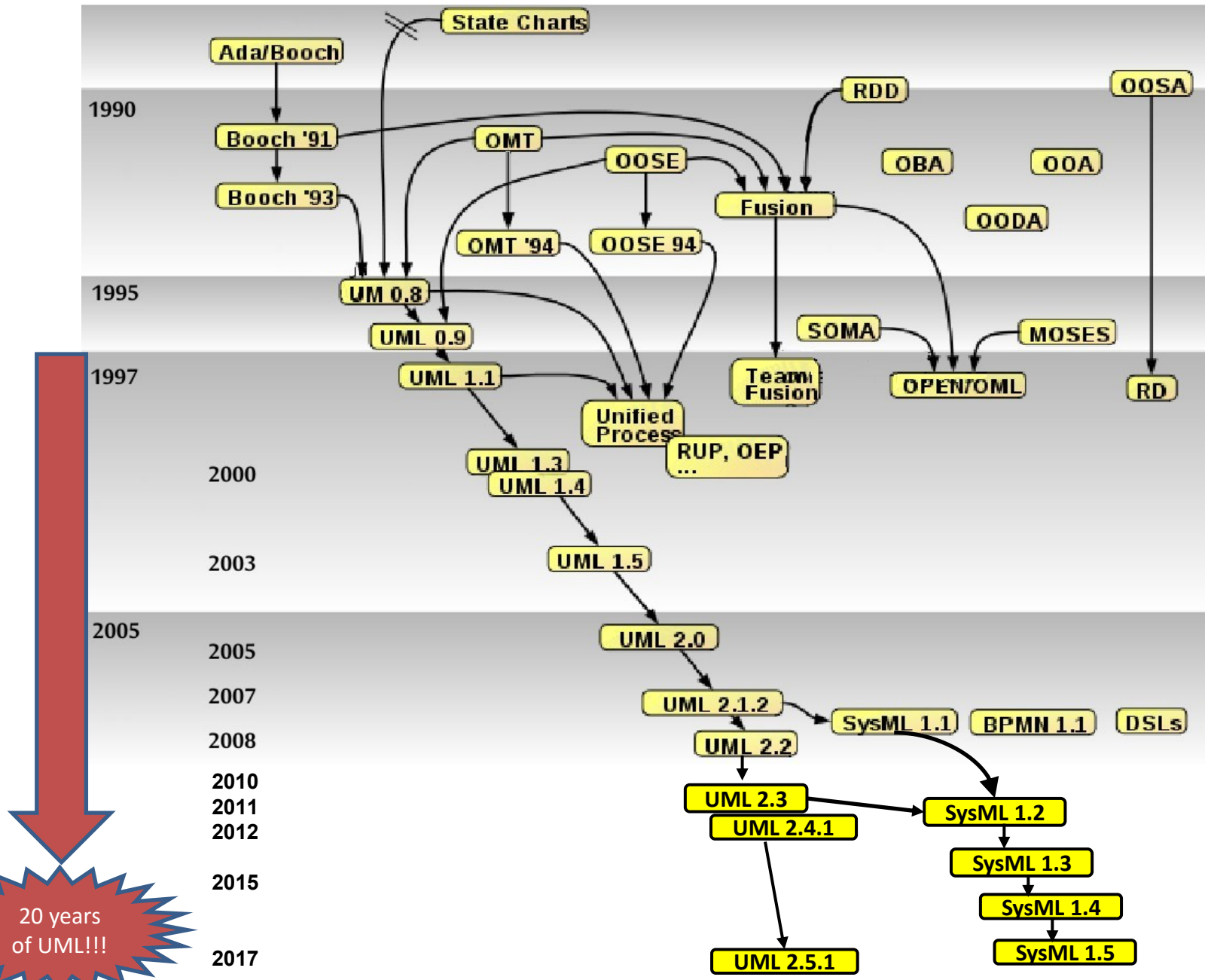
SysML



Event-based

Message-based

The path towards UML and SysML:



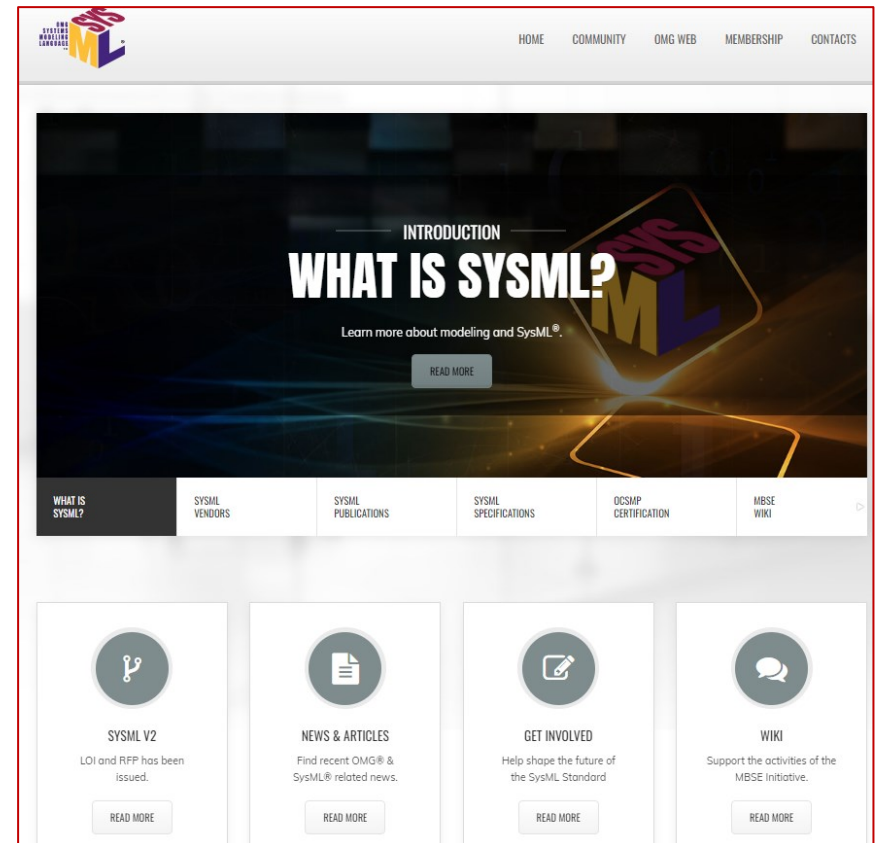
Unified Modeling Language

- A general-purpose modelling language focused on supporting software engineering concerns.
- Proposes a graphic notation to create visual models of software-intensive systems.
- Strongly influenced by object orientation.

<http://www.uml.org>

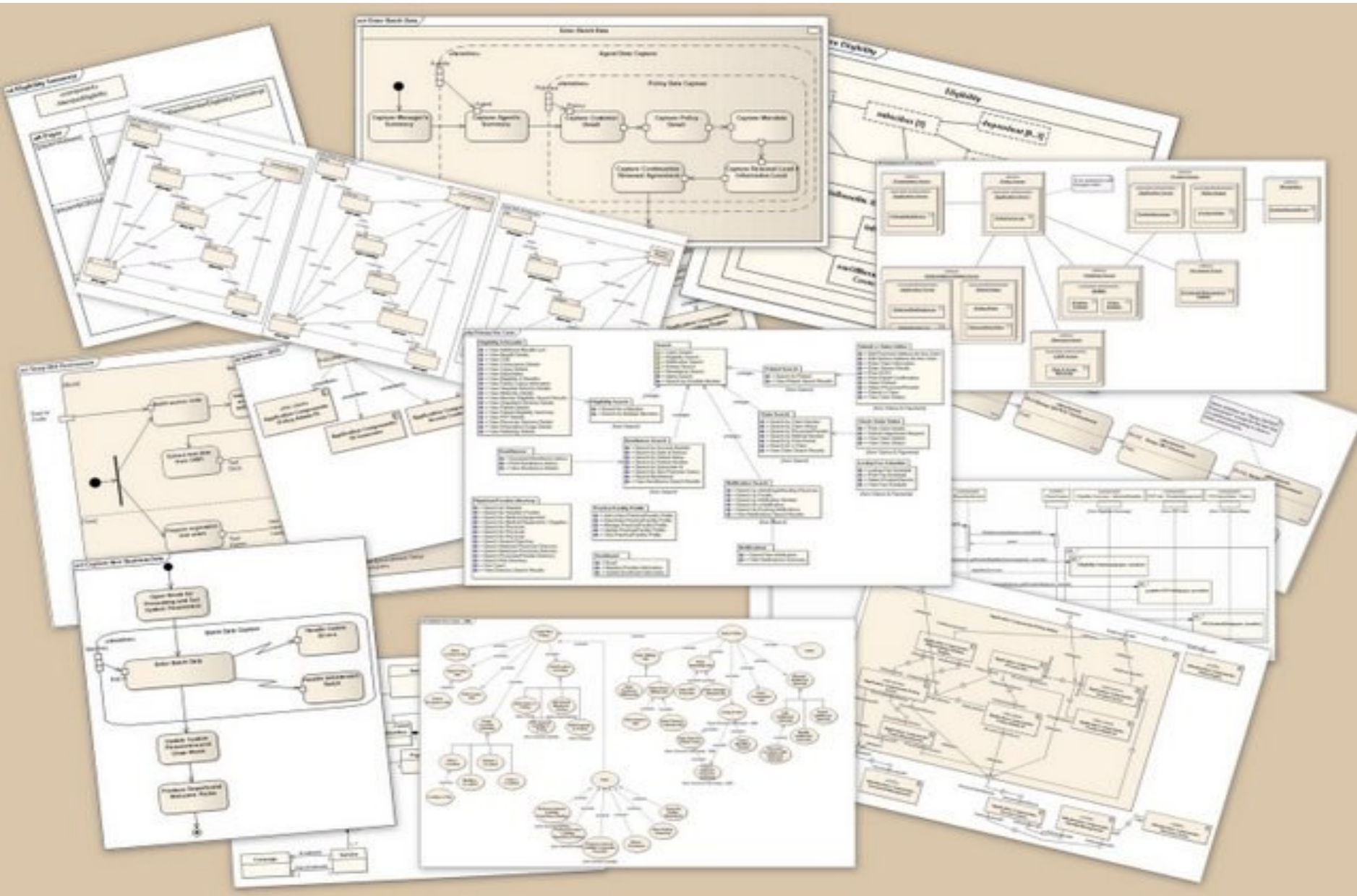


<http://www.omg.sysml.org>

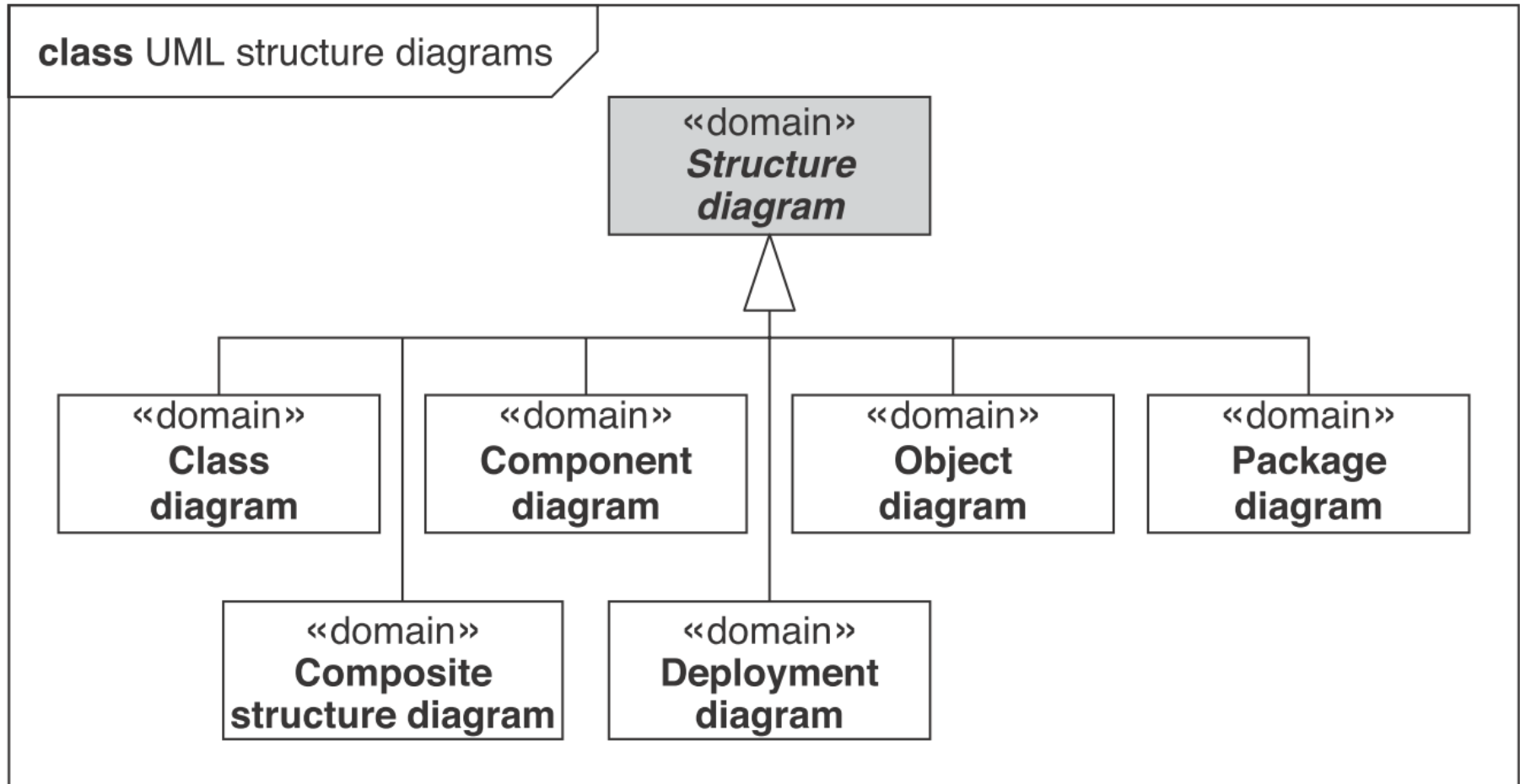


- The UML 2.x specification has four parts.
 - **Superstructure** defines the notation and semantics for diagrams and their model elements.
 - **Infrastructure** defines the core metamodel on which the UML Superstructure is based.
 - **Object Constraint Language (OCL)** for defining rules for model elements.
 - **UML Diagram Interchange** that defines how diagram layouts are exchanged.

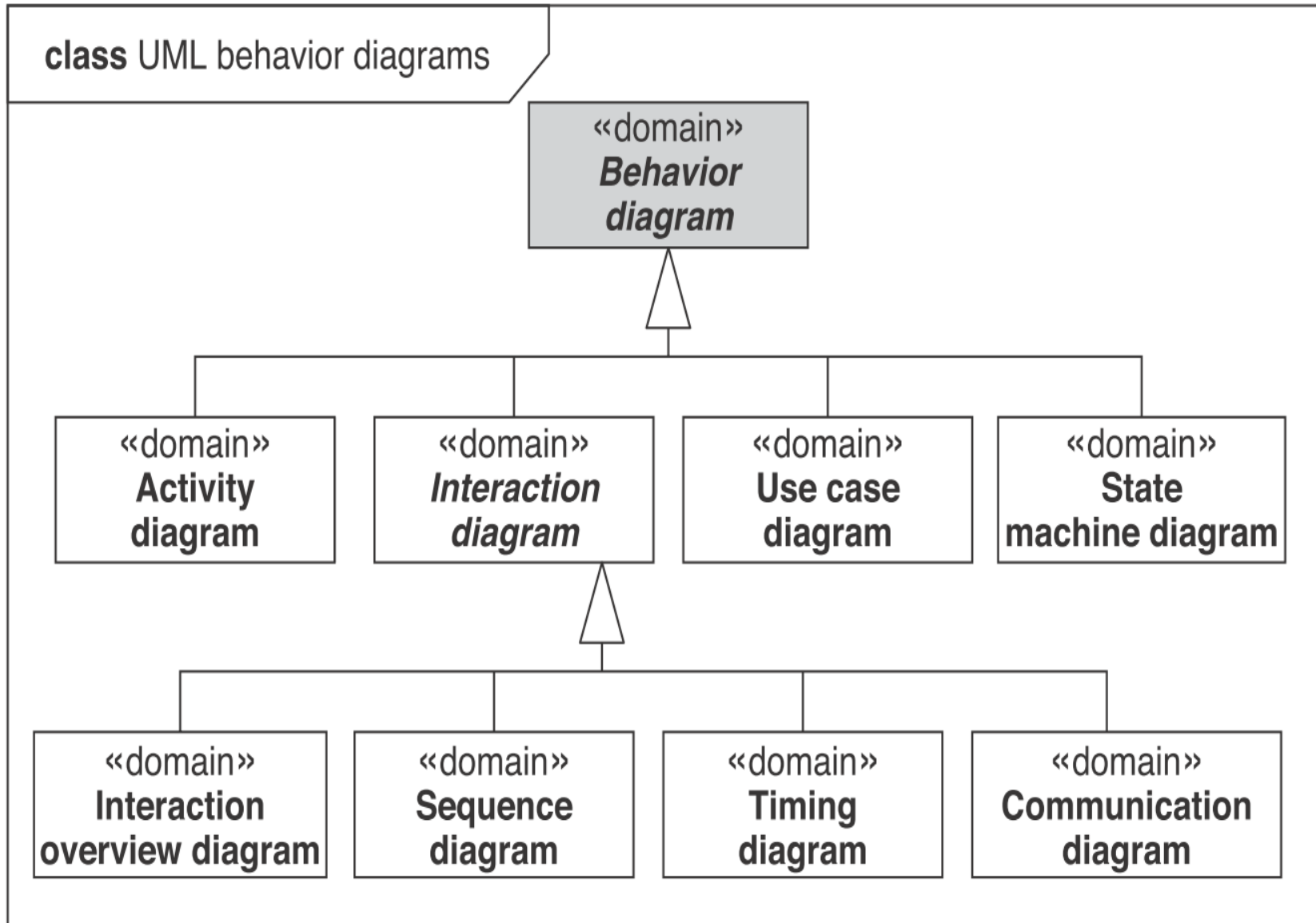
Views & Viewpoints in UML 2.x



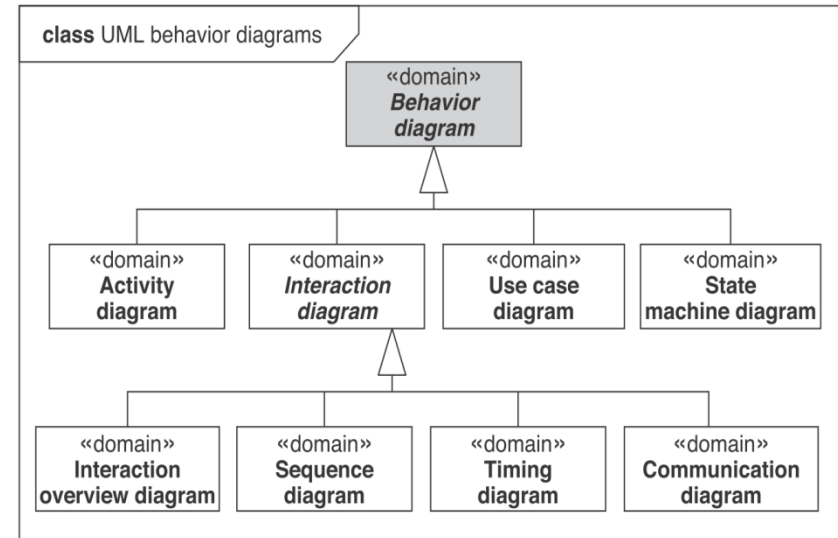
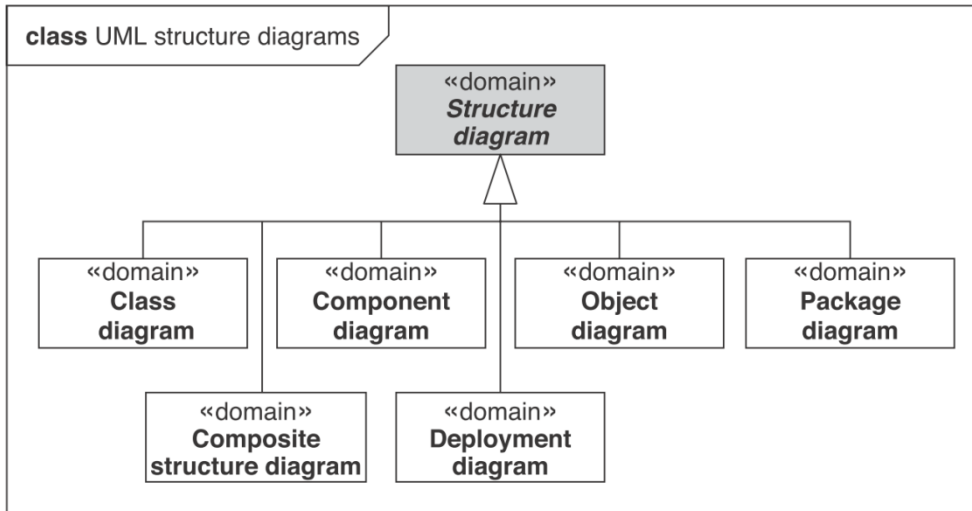
UML 2.x structure diagrams



UML 2.x behaviour diagrams



All UML 2.x diagrams

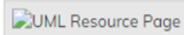


[CERTIFICATION](#)[TRAINING](#)[RESOURCES](#)[VENDORS](#)

UML RESOURCES



USEFUL LINKS TO UML RESOURCES

[UML Resource Page](#)

If you have, or know of, a link that we should add to these lists, send an email to [Dr. Jon Siegel](#), OMG's Vice President, Technology Transfer at siegel AT omg.org.

OTHER UML RESOURCES:


- [Introduction to UML](#)
- [Conrad Bock's UML 2.0 articles](#)
- [No Magic – MagicDraw UML](#)
- [Sparx Systems' UML Resource Page](#)
- [Sparx Systems Interactive Representation of UML](#)

UML PROFESSIONAL CERTIFICATION:


- [OMG'S UML Professional Certification Program](#)

UML TUTORIALS:

- [No Magic - MagicDraw UML](#)
- [OMG'S List of Training in OMG Technologies including UML](#)
- [Sparx Systems' UML 2.0 Tutorial](#)
- [Introduction to UML: Structural Modeling and Use Cases
Lecture 1 as omg/2001-03-02 by Cris Kobryn.](#)
- ["Behavioral Modelina"](#)



Central Europe



UML-Training | Registered users | FAQ | BLOG | Newsletter Registration | Support | Contact | About Us

- UML 2.4.1 Metamodel
 - 1. Start
 - Start
 - 2. UML Normative Views
 - UML Normative Views
 - 2.1 Actions
 - Actions
 - 2.1.1 BasicActions
 - BasicActions
 - 2.2 StateMachines
 - StateMachines
 - 2.2.1 BehaviorStateMachines
 - BehaviorStateMachines
 - Events
 - 2.2.2 ProtocolStateMachines
 - ProtocolStateMachines
 - 2.3 UseCases
 - UseCases
 - 2.4 Components
 - Components
 - 2.4.1 BasicComponents
 - BasicComponents
 - 2.4.2 PackagingComponents
 - PackagingComponents
 - 2.5 Activities
 - Activities
 - 2.5.1 IntermediateActivities
 - IntermediateActivities
 - 2.5.2 ExtraStructuredActivities
 - ExtraStructuredActivities
 - 2.5.3 BasicActivities
 - BasicActivities
 - 2.5.4 FundamentalActivities
 - FundamentalActivities
 - 2.5.5 CompleteStructuredActivity
 - CompleteStructuredActivities
 - 2.5.6 CompleteActivities
 - CompleteActivities
 - 2.5.7 StructuredActivities

UML 2.4.1 Metamodel


Feedback

1. Start


Feedback

Start

Feedback




OBJECT MANAGEMENT GROUP
www.omg.org



UML 2.4.1

UNIFIED
MODELING
LANGUAGE



www.uml.org

This Project contains a merged version of the UML Metamodel. Hence, the metaclasses are located in one single package. This prevents the need to describe the same metaclasses multiple times.


The original model (XMI) doesn't contain diagrams. Therefore, we have added diagrams to express different parts of the UML metamodel. For a better understanding, we have added some examples which describe the metamodel.

www.omg.org/spec/UML/20110701/UML.xmi

UML 2.4.1 Metamodel Classes

notes

This package contains all UML metamodel classes, but no diagrams.



Document Version 1.2.1

UML Normative Views

- + Actions
- + Activities
- + Components
- + StateMachines
- + UseCases

notes

This package shows the package structure of UML. Each package contains a diagram with the elements contained in this package. Because this is the merged version of UML, the element are not contained within the packages. They are located under the UML Package.

Diagrams

Primitive Links

- + Transition
- + Activity Edge
- + Association Example
- + Communication Path
- + Connector
- + Dependency
- + Extend
- + Include
- + Generalization
- + InformationFlow
- + ActivityEdge

notes

Shows an excerpt of the metamodel for UML-Links.

Views

- + Action Kinds
- + Activity
- + Behaviors
- + Classifiers
- + Constraint
- + Deployment
- + Different Kinds of Actions
- + Interaction Example
- + Interactions
- + Interface
- + Modeling Stack
- + Nodes
- + Operation
- + Package
- + Pin
- + Ports
- + Primitive Types
- + Templates
- + Time

- In January 2001 the International Council on Systems Engineering (INCOSE) adopted **UML** as a language for systems engineering applications.
- UML was then **adapted and extended** to systems engineering using **UML Profiles**.
- The **Systems Modelling Language (SysML 1.0)** was approved by the OMG in July 2006
 - based on UML 2.1.1 ...

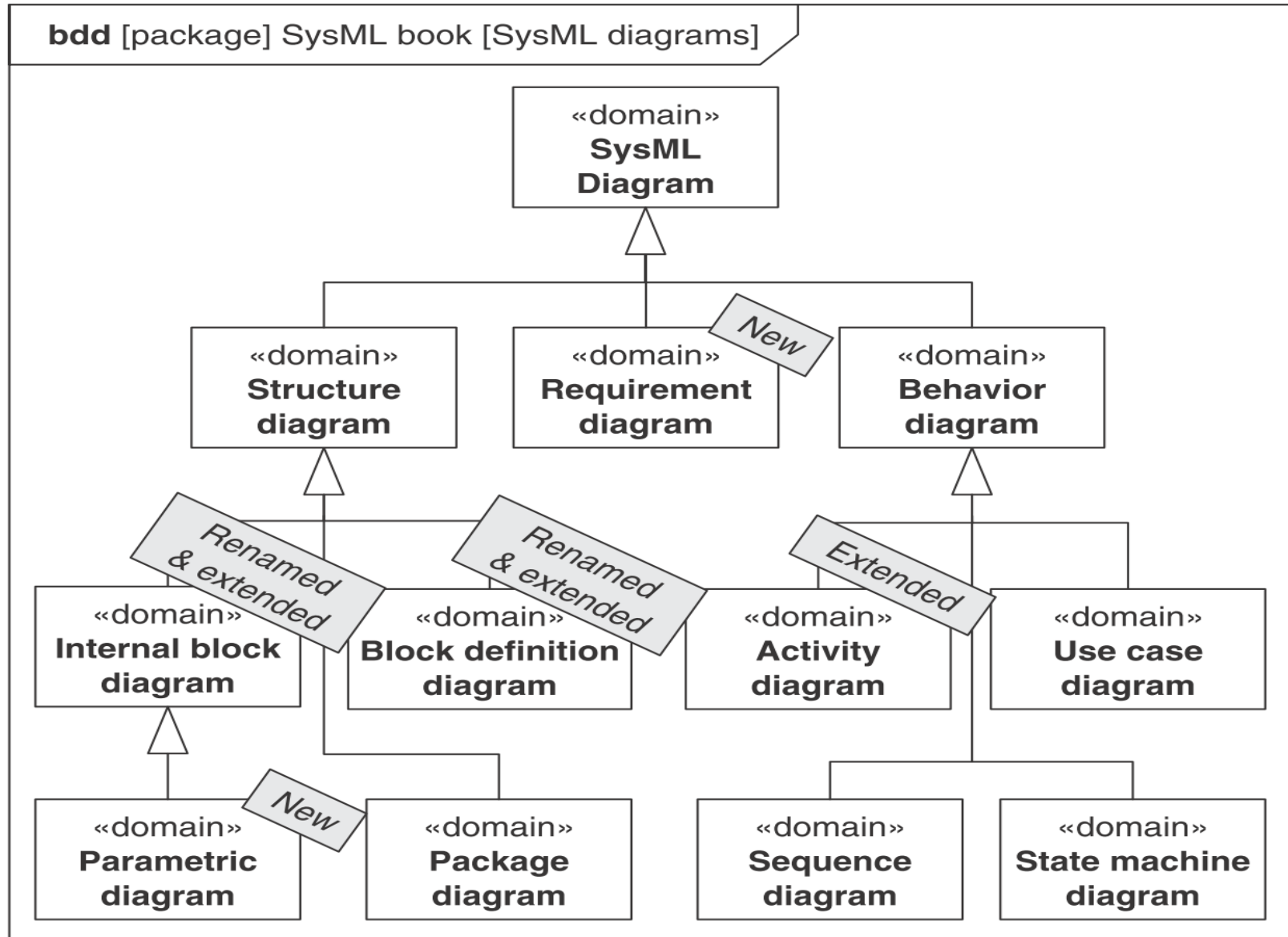
Why SysML? Is UML not enough?

- UML lacks **requirements modelling**.
 - Whereas requirements modelling is an explicit component of systems engineering (and thus is part of SysML).
- UML has a rather **software and information-specific** scope as it focus on modelling **logical** systems.
 - SysML as a wider **system-specific** scope enabling it to model both *logical* and *physical* systems.
- UML is strongly influenced by **object orientation**.
 - SysML is neutral regarding the abstraction paradigm.

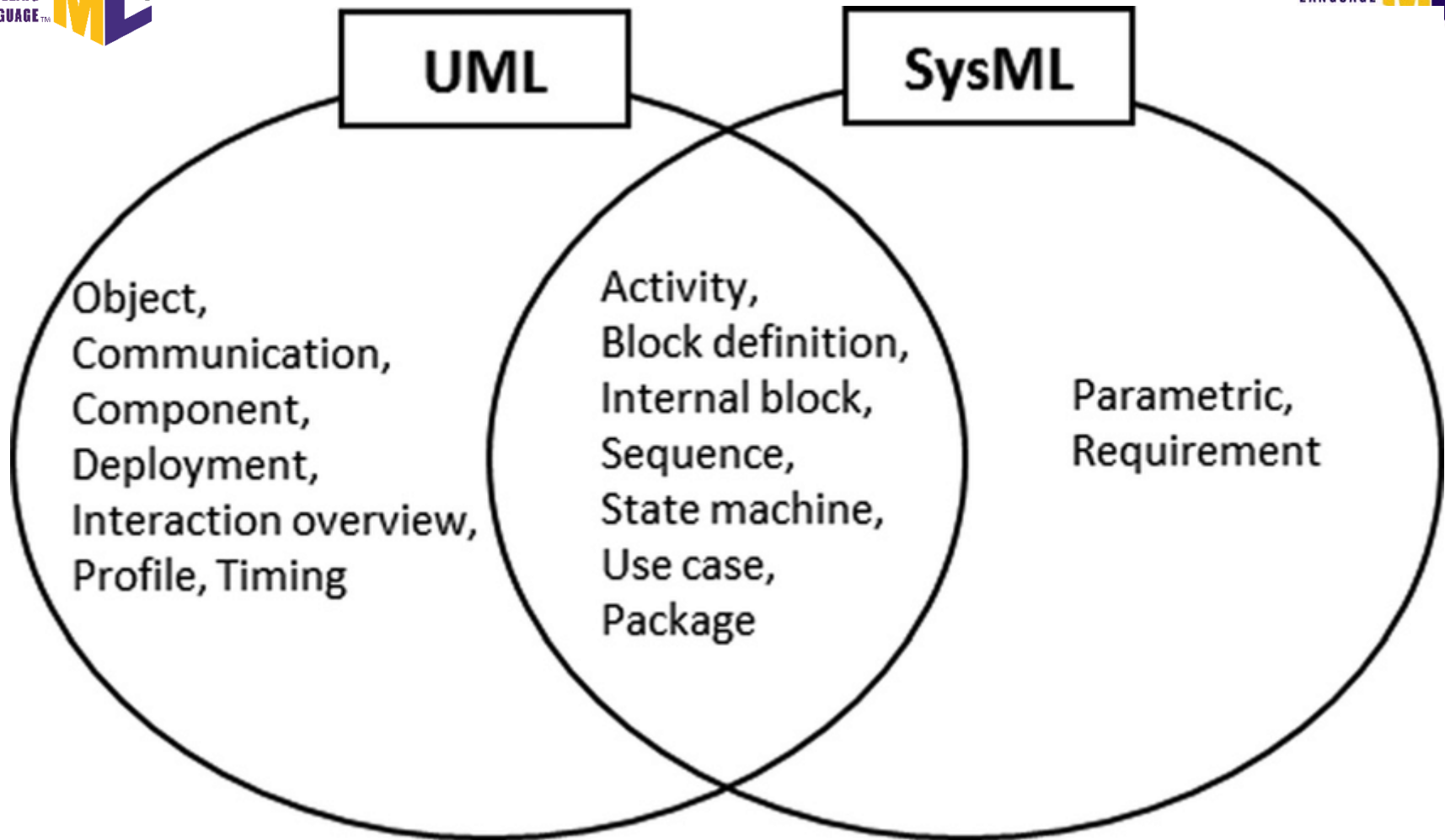
- “**Logical**” **systems** are typically realized as **software artefacts**.
 - Focus on “**concept**” representation.
 - These concepts can be modelled using object orientation principles such as abstraction, classification and generalization.
 - **UML** specializes in modelling these systems.

- “**Physical**” **systems** are typically realized as **tangible** artefacts, such as devices, machines and hardware.
 - Focus on “**object/part**” representation alongside with concepts.
 - **SysML** specializes in modelling these systems...

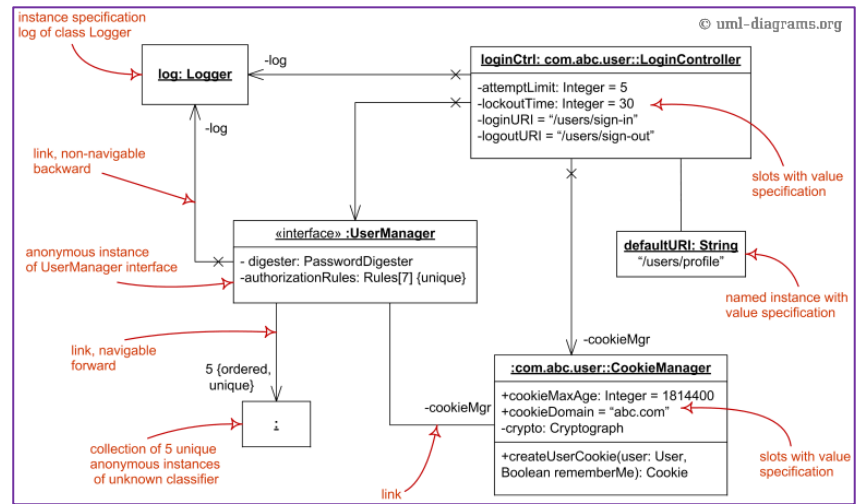
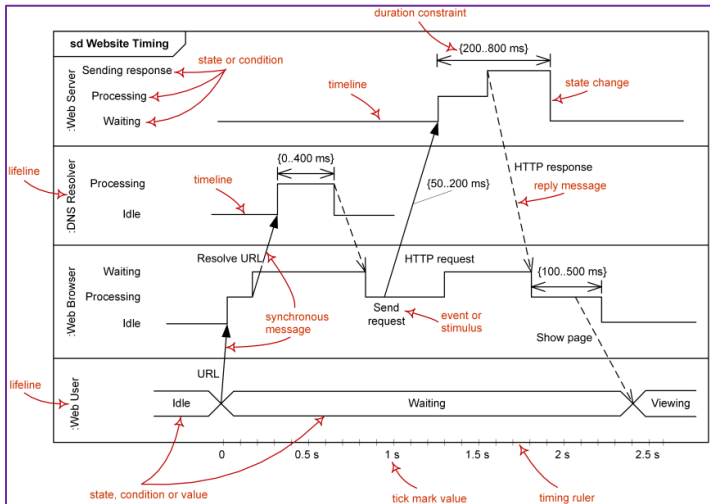
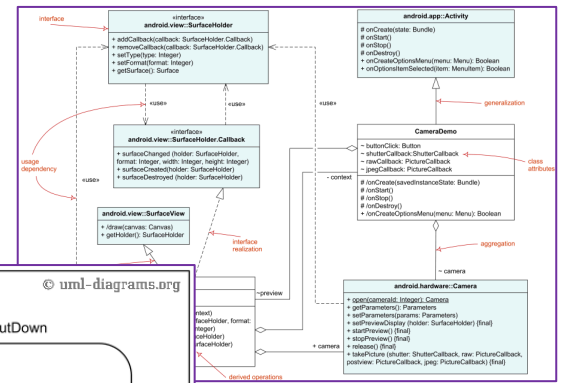
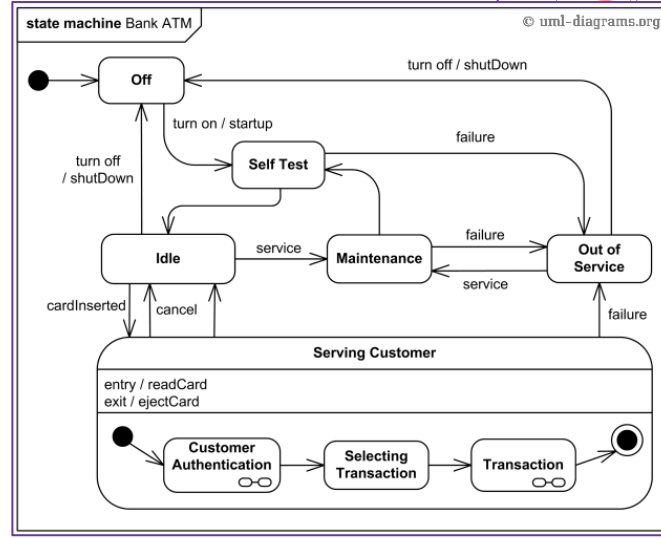
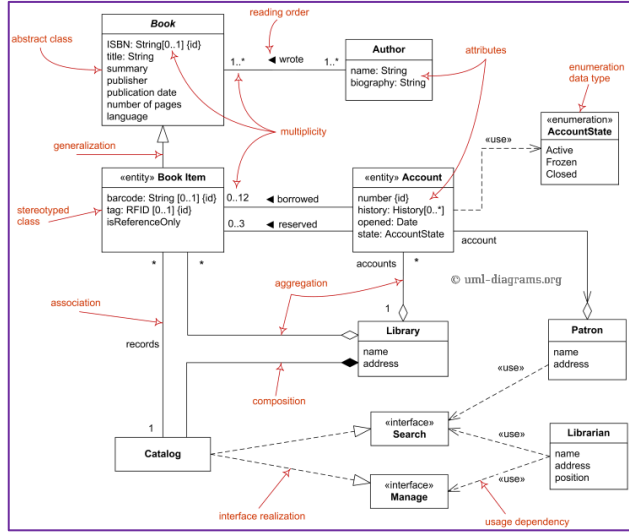
SysML diagrams (comparing with UML)



UML and SysML diagrams



Many examples and clues at <https://www.uml-diagrams.org/>



UML and SysML as ISO standards...

- UML 2.4.1 is ISO/IEC 19505:2012
 - <https://www.iso.org/standard/32624.html>
 - <https://www.iso.org/standard/52854.html>
- SysML 1.4 is ISO 19514:2017
 - <https://www.iso.org/standard/65231.html>

Home > Store > Standards catalogue > Browse by ICS > 35 > 35.060 > ISO/IEC 19505-1:2012

ISO/IEC 19505-1:2012

Preview

Information technology -- Object Management Group Unified Modeling Language (OMG UML) -- Part 1: Infrastructure



This standard was last reviewed and confirmed in 2017. Therefore this version remains current.

ISO/IEC 19505-1:2012 defines the Unified Modeling Language (UML), revision 2. The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.

Home > Store > Standards catalogue > Browse by ICS > 35 > 35.060 > ISO/IEC 19505-2:2012

ISO/IEC 19505-2:2012

Preview

Information technology -- Object Management Group Unified Modeling Language (OMG UML) -- Part 2: Superstructure



This standard was last reviewed and confirmed in 2017. Therefore this version remains current.

ISO/IEC 19505-2:2012 defines the Unified Modeling Language (UML), revision 2. The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.

Home > Store > Standards catalogue > Browse by ICS > 35 > 35.060 > ISO/IEC 19514:2017

ISO/IEC 19514:2017

Preview

Information technology -- Object management group systems modeling language (OMG SysML)

The purpose of ISO/IEC 19514:2017 is to specify the Systems Modeling Language (SysML), a general-purpose modeling language for systems engineering. Its intent is to specify the language so that systems engineering modelers may learn to apply and use SysML; modeling tool vendors may implement and support SysML; and both can provide feedback to improve future versions. Note that a definition of "system" and "systems engineering" can be found in ISO/IEC/IEEE 15288.

SysML reuses a subset of UML 2 and provides additional extensions to satisfy the requirements of the language. This International Standard documents the language architecture in terms of the parts of UML 2 that are reused and the extensions to UML 2. The International Standard includes the concrete syntax (notation) for the complete language and specifies the extensions to UML 2. The reusable portion of the UML 2 standard is not included directly in the International Standard.