Enterprise Architecture Languages

Business Process Modeling Notation (BPMN)

- This was originally developed by the *Business Process Management Initiative (BPMI)*. In 2005, the Object Management Group took over the initiative (Lucid Software Inc., n.d.).
- This is also known as *Business Process Model and Notation*.
- The goal of BPMN is to support business process modeling by providing a comprehensive standard notation to business users, while representing the complex process semantics for technical users.
- It is intended to be used by stakeholders who design, manage, and realize business processes (BP).
- It has an easy to use flowchart-like notation that is independent of any particular implementation environment.
- It creates a standardized bridge for the gap between the business process design and process implementation (Object Management Group, n.d.).
- This helps with communication leading to Extensible Markup Language (XML).

Element Types in Business Process Diagrams (Lucid Software Inc., n.d.)

- **Flow Objects**
  - **Event** – A trigger that starts, modifies, or completes a process. Event types include message, timer, error, compensation, signal, cancel, escalation, link, etc. An event can be classified as either *"throwing"* or *"catching,"* depending on their function.

Figure 1. Notation of events in BPMN.
Source: https://www.lucidchart.com/pages/bpmn

  - **Activity** – A particular task performed by a person or system. It can include sub-processes, loops, multiple loops, and compensations.

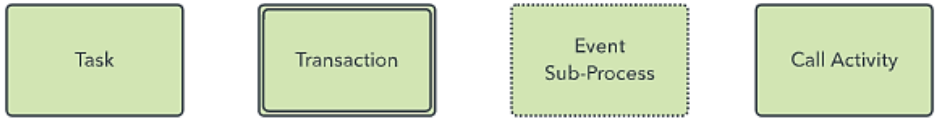

Figure 2. Notation of different activities in BPMN.
Source: https://www.lucidchart.com/pages/bpmn

  - **Gateway** – A decision point that can adjust the path based on conditions or events. A gateway can be exclusive or inclusive, parallel, complex, or based on data or event.

Figure 3. Notation of gateways in BPMN.
Source: https://www.lucidchart.com/pages/bpmn

- **Connecting Objects**
  - **Sequence flow** – This shows the order of activities to be performed.
  - **Message flow** – This depicts messages that flow across pools, or organizational boundaries such as departments. It should not connect events or activities within a pool.
  - **Association** – This associates an artifact or text to an event, activity, or a gateway.
- **Swimlanes**
  - **Pool** – It represents major participants in a process. A different pool may be in a different company or department but still involved in the process.
  - **Lane** – It is a sub-part of a pool. It shows the activities and flow for a certain role or participant, defining the accountability within the processes.

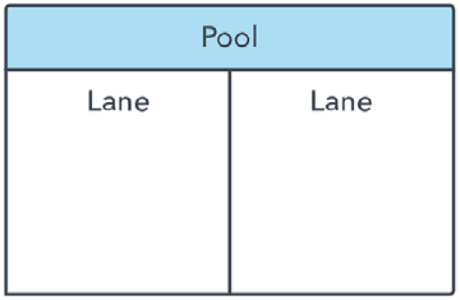

Figure 4. Notation of swimlanes in BPMN.
Source: https://www.lucidchart.com/pages/bpmn

- **Artifacts**
  - **Data object** – This shows the necessary data for an activity.
  - **Group** – This shows a logical grouping of activities, but does not change the diagram's flow.

- **Annotation** – This provides further explanation to a part of the diagram.

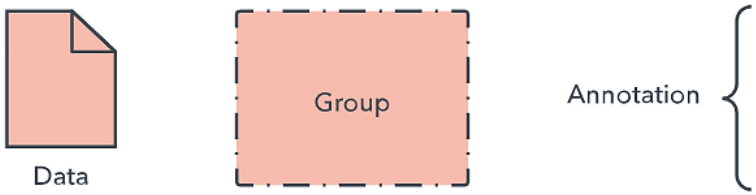

Figure 5. Notation of artifacts in BPMN.
Source: https://www.lucidchart.com/pages/bpmn

Sub-Models Within A BPMN Diagram (Lucid Software Inc., n.d.)

- **Private BP** – This involves internal processes to a specific organization and do not cross pools or organizational boundaries.
- **Abstract BP** – This occurs between a private BP and an external participant or process. This does not show the private BP itself.
- **Collaborative BP** –This shows the interactions between two or more business entities.

Unified Modeling Language (UML)

- This was created to establish a common, semantic, and syntactical visual modeling language for the architecture, design, and implementation of complex software systems both structurally and behaviorally (Lucid Software Inc., n.d.).
- This is currently the most important industry-standard language for specifying, visualizing, constructing, and documenting software systems.
- It emerged from the combination of three (3) existing practices namely, the Booch Method, Object-Modeling Technique (OMT), and Objectory (Lankhorst, 2017).
- This is commonly used by programmers, but not generally used by database developers.
- The Object Management Group (OMG) oversees the definition and maintenance of the UML specifications, which provides engineers and programmers the ability to use one language for different purposes during all phases of the software lifecycle, for all system sizes (Lucid Software Inc., n.d.).

Modeling Concepts Specified by UML (Lucid Software Inc., n.d.)

- **Functional** – This concept involves use case diagrams, which describes system functionality from the point of view of a user.
- **Object** – This concept involves class diagrams, which describes the structure of a system in terms of objects, attributes, associations, and operations.
- **Dynamic** – This concept involves interaction diagrams, state machine diagrams, and activity diagrams, which are used to describe internal behavior of the system.

Types of UML Diagrams (SmartDraw, LLC, n.d.)

- **Structural UML Diagrams**
  - **Class diagram** – It describes the static structure of a system.
  - **Package diagram** – It is a subset of a class diagram used to organize elements of a system into related groups.
  - **Object diagram** – It describes the static structure of a system at a particular time, and can be used to test class diagrams for accuracy.
  - **Component diagram** – This describes the organization of physical software components, including source codes, run-time code, and executables.
  - **Composite structure diagram** – This diagram shows the internal parts of a class.
  - **Deployment diagram** – This depicts the physical resources in a system, including nodes, components, and connections.
- **Behavioral UML Diagrams**
  - **Activity diagram** – This illustrates the dynamic nature of a system by modeling the flow of control from activity to activity.
  - **Communication diagram** – This describes the interactions among classes in terms of an exchange of messages over time.
  - **Use case diagram** – This models the functionalities of a system using actors and use cases, and can be considered as a simplified version of a collaboration diagram introduced in UML 2.0.
  - **State machine diagram** – This describes the dynamic behavior of a system in response to external stimuli.
  - **Sequence diagram** – It models the interactions between objects in sequence.
  - **Interaction overview diagram** – It is a combination of an activity and a sequence diagram, which models a more complex interaction.
  - **Timing diagram** – It is an interaction UML diagram that focuses on processes that take place during a specific period of time, wherein time is shown to increase from left to right.

Architecture Description Language (ADL)

- It is a general term referring to different formal languages used in enabling formalization, description, specification, modeling and reasoning of software architectures (Vac, 2017).
- A good ADL must be capable of providing adequate abstractions in modeling large systems.
- A wide variety of ADLs exist, with several differences in the exact concepts it offer, and are mainly suitable for users with technical background (Lankhorst, 2017).
- Some examples of ADLs are UniCon, Wright, and Acme.

Important Properties of ADLs

- Ability to represent components along with property assertions, interfaces, and implementations
- Ability to represent connectors along with protocols, property assertions, and implementations
- Component and communication abstraction and encapsulation
- Ability to accommodate analysis tools openly
- Limiting communication to those components connected to each other architecturally
- Ability to model dynamic architectures
- Ability to reason about causality and time
- Hierarchical refinement support
- Ability to check conformance

References:

Lankhorst, M. et al. (2017). *Enterprise Architecture at Work Modeling, Communication and Analysis* (4th ed.). Berlin, Germany: Springer Nature

Lucid Software Inc. (n.d.). *What is Business Process Modeling Notation?*. Retrieved from https://www.lucidchart.com/pages/bpmn on December 17, 2019

Lucid Software Inc. (n.d.). *What is Unified Modeling Language?*. Retrieved from https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language on December 18, 2019

Object Management Group. (n.d.). *Graphical Notations for Business Processes*. Retrieved from https://www.omg.org/bpmn/ on December 17, 2019

SmartDraw, LLC. (n.d.). *UML Diagram*. Retrieved from https://www.smartdraw.com/uml-diagram/ on December 18, 2019

Vac, G. (2017). Architecture Description Languages. *Today Software Magazine Issue 55*. Retrieved from https://www.todaysoftmag.com/article/2241/architecture-description-languages on December 19, 2019