

Unified Modeling Language

Engineering is all about developing solutions by reducing complex reality to tractable bits and pieces. At the heart of problem solving lies modelling, that is, isolating the essential aspects of the domain from the rest of the complicated real world. In course of time many tools have been developed to assist the engineer in modelling real-world processes. Today many such processes are captured by software (IT) solutions running on worldwide, interconnected computers. Indeed, information society is progressing at such a pace that in many countries today you can book a flight, retrieve money from the bank, make an appointment for a visit to the doctor and even - however reluctantly - authorise payment to the furniture vendor who has begun legal proceedings against you. All along IT lines.

Odd Three-letter

But how to warrant that during design, development, implementation and maintenance of automated processes, business analysts, designers and programmers all speak a common vocabulary? In the past decade UML (Unified Modeling Language) has emerged to act as such a universal language. It has been developed first and foremost as a process and tool-independent modelling syntax for building software systems. However, UML can be used for various other application areas. For example, in this issue of GIM International RadoÅi Åumrada presents the use of UML for modelling the process of transferring real estate from seller to buyer in Slovenia. For those to whom UML is still an odd three-letter abbreviation, here is some basic background.

Graphical Notation

When one wants to (partially) convert a process (dynamic system) taking place in, for example, the cadastral domain - such as registration of a transfer act within a land administration system - into a software system, the process may be approached from several directions.

- What is the user's view on the process?
- What are the objects in the system and how do they interrelate? (Static system view).
- Who are the actors and what activities do they perform?
- At what point do actors come into action, in what sequence and for how long?
- How do the actors collaborate?
- li>Which software components are involved in running the process on a hardware system?

UML (Unified Modeling Language) has been developed to describe the different views on a domain process in graphical notations in the form of diagrams.

No less than nine modelling diagrams are distinguished:

- use case diagram
- class (package) diagram
- object diagram
- sequence diagram
- collaboration diagram
- statechart diagram
- activity diagram
- component diagram
- deployment diagram.

Stick Figures

A UML user will not commonly employ the entire set of above diagrams but will make a selection according to needs and the role he plays in the modelling train moving from real-world process to software system. For example, a designer may restrict himself mainly to the use of class diagrams because these form a pivotal link between the real-world domain process and its transformation into a software system running on hardware. The analyst, concerned with understanding what is actually happening within the domain process, will generally prefer modelling the problem along the lines, ovals and stick figures used in case diagrams.

IT Tough Guys

Sequence diagrams enable modelling of time within the process, while activity diagrams show actions carried out by objects and components. Just as the "use case diagram" comes closest to the real-world process, component and deployment diagrams are nearest to the software system: component diagrams show the types of software elements, their interfaces and dependencies, while deployment diagrams represent the physical configurations of software and hardware, including nodes, links and dependencies. These diagrams are mainly of interest to the IT tough guys - software programmers and hardware assemblers who keep the system running, possibly Web, but at least server-based. For further details on the various UML diagrams see the brief introduction presented in the text box in Sumrada's article.

Selling Point

Within the geo-IT domain UML is gaining momentum because it supports the interoperability of geo-information. This is important for dissemination of geo-information within the framework of establishing a National Geo-information Infrastructure. Today most standards such OGC, ISO TC211, and CEN, are even written in UML. So the system developer who represents his design in accordance with UML syntax is developing along standardised lines, which is in itself a sound selling point.

<https://www.gim-international.com/content/article/unified-modeling-language>
