Applications of Implementation Diagrams in System Infrastructure Modeling

Introduction

Information system (IS) modeling domain is commonly associated with visual representation of the functionalities of system under development [comp. Shoval and Kabeli, 2005]. It concerns in particular the specification of functional requirements – and less frequently – non-functional ones, due to the lack of the commonly accepted system requirement modeling standard for the latter [Zou and Pavlovski, 2008]. Requirements regarding IS services are defined basically on a conceptual level. Nevertheless, in order to carry out the implementation phase of the system lifecycle properly, laying out technical aspects, related to IS hardware and software, is clearly necessary. Architecture-related aspects of information system – both software and hardware – are visualized with UML using a number of strictly connected diagrams, i.e. component and deployment diagrams. These UML constructs are commonly grouped within dedicated architecture views [comp. Kruchten, 1995]. Due to substantial content-related relationships as well as application practice, authors of the current article address aforementioned pair of diagrams using an abstract notion of implementation diagrams (see figure 1).

Although MIS is one of the most quickly evolving areas of research and business activity, UML – including its implementation features – retains its exceptional stability, despite its relatively long history. UML still meets the crucial expectations of IS analysts and designers regarding the most current and modern digital and mobile applications as well. Despite UML stability, still new current research publications appear with the aim
of solving specific complex issues, such as UML-RT: Real-Time Systems applications [Bianco, Lavazza, Mauri and Occorso, 2003] or data warehouse architecture modeling [Lujan-Mora, Trujillo and Song, 2006]. The goal of the paper is to assess as well as to verify challenges and opportunities related to the applications of implementation diagrams within the development process of modern digital economy-oriented systems. In order to achieve that, literature/documentation analysis was conducted and supported by a series of implementation-oriented case studies. The paper covers also principles of the effective applications of component and deployment diagrams.

1. The role of implementation diagrams in infrastructure modeling

Both UML – Unified Modeling Language [Booch, Rumbaugh and Jacobson, 2004], [Object Management Group, 2012] – as well as BPMN – Business Process Model and Notation [Object Management Group, 2013] became the vital standards used in information systems development domain. The standards are commonly used in business applications and academic research/teaching [Wrycza and Marcinkowski, 2005]. The particular attention is paid to UML, with its widespread diagrams used to specify functional requirements (represented by use cases), classes, flows of activities and interactions within information system.

Practically, UML usage relies on the proper selection and application of the most suited diagrams to the specific information system functionality and/or structure aspects. A rich set of 14 types of the diagrams is available to a system designer within the process of information system modeling. Usually, due to the UML complexity, the four most accepted and widespread types of diagrams are integrated and addressed as so called light or minimal versions of UML superstructure [Wrycza and Marcinkowski, 2007]. The similar outcome resulted from the parallel research carried out by [Erickson and Siau, 2007].

On the other hand, a number of UML constructs and diagrams have been quite often underestimated or neglected. System analysts and designers usually regard these types of diagrams as playing the supplementary or highly specialized role in the IS development process. That is the case regarding implementation diagrams [Al-Fedaghi and Abdullah, 2013]. However, it ought to be recognized as one of the major UML strengths that information systems designers are enabled to model vari-
ous architectural perspectives, including physical system infrastructure comprising computer and networking hardware as well as software [Holt, 2007]. Having such a robust functionality, UML standard meets major quality metrics required from any system- or business-oriented modeling notation, such as expressiveness, suitability, completeness or effectiveness [comp. Hommes and van Reijswoud, 2000]. The emplacement of implementation diagrams in the UML diagrams hierarchy is presented at figure 1.

Figure 1. The emplacement of the implementation diagrams in diagrams hierarchy for UML 2.4.1

2. Implementation diagrams – functional capabilities

In accordance with the leading system lifecycle patterns – waterfall, spiral and iterative-incremental ones – both functional and non-functional aspects of the system modelled should be identified to enable efficient analysis, design as well as implementation of software and hardware design. As shown at figure 1, the abstract notion of implementation diagrams comprises component diagrams and deployment diagrams. The software-related implementation-oriented perspective is modelled using component diagrams, while the hardware-related one – by deployment diagrams.
Component diagram is a kind of implementation diagram that is oriented on description of the organization of software components and relationships among them. The basic conceptual and notational categories of components diagrams include a component, required interface, provided interface, port as well as complex port. In respect of relationships – system analysts are enabled to use dependencies, realizations, delegate connectors and assembly connectors. Interfaces lay out sets of operations that point out the services offered by a class or a component. Any component may be replaced freely by another component as long as their provided and required interfaces are compatible. Practically, the basic kinds of components are first and foremost: executable programs, program libraries, databases, flat files, data tables, subsystems or system services [Ahmed and Umrysh, 2002].

The set of concepts utilized within deployment diagrams is considerably more limited compared to component diagrams. In turn, deployment diagram is a kind of implementation diagram which describes the network of nodes interconnected with communication paths, along with the artifacts allocated to relevant nodes. The nodes represent basically computer and networking hardware. Specifying the components and other IT artifacts that reside on individual nodes is not mandatory. In comparison to component diagrams – basic terminology of deployment diagrams covers only nodes and communication paths as a relationship type between nodes. Thus, a node is a physical or logical processing resource that enables executing components and utilizing artifacts allocated to the node. To provide some examples, general ICT equipment and execution environments such as operating systems, e-learning platforms (Moodle, BlackBoard, WebCity, Sakai etc.), ERP, CRM, WFM systems, database management systems, e-shops, e-auctions, conference management systems are all classified as system nodes.

3. Implementation diagrams in digital economy – practical applications

Theoretical aspects of implementation diagrams may be exemplified using diverse practical examples. In order to achieve that goal, the following applications originating from the digital economy domain were selected, analyzed and discussed:

– component diagram – intelligent system for metropolitan traffic management,
deployment diagram – information system for the store service,
deployment diagram that integrates relevant elements of component
diagram – mobile devices-based electronic ticket purchase.

The complex aspects of software modeling were the matter of detailed studies and analysis, specific for the above mentioned examples of information systems, eg. [Larman, 2012; Mahmood and Lai, 2008; Mokarrrat and Vatanawood, 2013; Wrycza, Marcinkowski and Wyrzykowski, 2006]. Due to the scope and goal of the paper, only selected results are included.

3.1. Intelligent system for metropolitan traffic management

The existing urban infrastructure within most metropolitan areas imposes strong limitations for the smooth flow of individual vehicles throughout road network, causing traffic jams. It is in particular bothersome during peak hours. From a certain point, increasing the throughput of distinct roads as well as increasing the density of existing road network has no effect on transit times regarding individual transport, because additional infrastructure supply meets with even more accelerated demand. In order to limit congestions, a number of physical and organizational solutions are used.

One of the most advanced remedies for the problem in question are intelligent systems for the metropolitan traffic management, classified generally as Intelligent Transport Systems (ITS). Their application enables an essential improvement of traffic conditions by advanced traffic lights control, dependent on traffic congestions. Having that said, capabilities of such systems are much more sublime and include such features as mechanism for traffic safety management, road accident-related notifications in harmony with metropolitan emergency systems and also LED screens that display important warnings information for drivers. Some ITS implementations are designed exclusively for public transport improvement.

The concept of ITS has been introduced in Gdańsk Metropolitan Area, Poland in the form of a TRISTAR system. Its main functionalities are presented at figure 2, using the basic conceptual categories of component diagrams. Complete functionality of TRISTAR functionalities is described in [Jamroz, Krystek and Kustra, 2007].
As shown at figure 2, it is the Traffic Monitoring and Supervision Subsystem that is the central component of ITS system. The subsystem collects data from a number of detectors. After processing, the data is used consequently by remaining five subsystems and so is the functionality of core component. Thus, subsidiary subsystems are connected using dependencies with the Traffic Monitoring and Supervision Subsystem.

3.2. Information system for the store service

It is an information system for the store service called e-store that is the quite frequent example of the modern information system used in digital economy. The digital store service systems are common in commerce, construction business and many other business sectors. E-store system assigns the tasks regarding store service by distributing proper instructions to employees responsible for store handling. After receipt of
instruction, employees have to accomplish the received instruction of goods relocating. Their task completion is recognized and appropriate system status is assigned when the bar code of the specific commodity is be scanned accordingly to the instruction content. This employee action of release or moving the commodities is recorded in the system database, as illustrated at figure 3. E-store system considerably improves operation of the large space stores, where tens of thousands of different commodities are stored.

**Figure 3. Deployment diagram for information system of the store service**

It is an Intelligent Store Server that is the crucial node of the e-store system. The server manages communication among other nodes using communication paths. Two of the remaining nodes are particularly significant to overall system operation – Database Server is responsible for the vast system data storage, while a (unspecified directly) number of Bar Code Scanner nodes provides functionality, necessary to register the accomplishment of the tasks scheduled earlier.

### 3.3. Mobile devices-based electronic ticket purchase

Nowadays mobile phones are more and more widely used for making payments regarding different digital services. One of business-rele-
vant examples of such applications is a mobile devices-based electronic
ticket purchase system. It may prove to be a useful tool inter alia for pur-
chasing tickets for the metropolitan public transport service. After logging
into the payment mobile application, the customer selects means of trans-
port (subway, tram, trolleybus, metropolitan railway, bus etc.) along
with route number as well as ticket type (day ticket, single ticket etc.).
A payment for the ticket is credited from customer’s pre-paid or current
account. In case ticket inspection occurs, customer is obliged to provide
ticket details on the screen of her/his mobile device. The data provided are
then verified by a ticket inspector.

Mobile device-based electronic ticket purchase system is modeled using
deployment diagram that integrates elements of component diagram (figure
4).

Figure 4. Deployment diagram for mobile device-based electronic ticket
purchase system

Source: Own elaboration.
The system is controlled by the node of Application Server, connected by communication paths with two functionally dependent nodes (i.e. Ticket Inspector Mobile Device and Passenger Mobile Device) as well as Database Server that hosts database storing data updated in a regular manner. Most of the components that are singled out within the diagram rely on a DB Management component that handles various queries throughout the application. Authentication Service is provided by separate AAA Server.

Conclusion

In order to put into practice the requirements of the system designers regarding quickly changing information and communication technologies, both kinds of implementation diagrams, i.e. component diagrams and deployment diagrams, were assessed and applied. The system architecture-related data contained in them has the substantial value for systems designers in face of quickly changing and upgrading information and communication technologies.

The case studies and elaborated models included in this paper led the authors to conclusion that Unified Modeling Language, besides supporting high-level modeling of processes and interactions, thoroughly enables and assists the development of IS architecture – regarding ICT equipment as well as software components of the information system. Research conducted indicates that digital economy-aimed component models are often overcomplicated and hence rejected by business stakeholders. Since features such as interfaces, ports or assembly connectors are of secondary values while supplementing business specifications with technological software overview, it is recommended to keep component models as simple as possible by relying on dependencies, should no in-depth implementation details be required by IT project development team. On the other hand, deployment diagram proved to be the leading UML diagram in terms of graphical stereotyping. Stakeholders’ groups representing different industries are usually very open to introducing industry-specific stereotypes for device nodes as well as communication paths – networking industry being one of the most evident examples.

UML language still retains its usefulness in modeling contemporary applications, taking into account quick technological development of ICT field. Its vitality is related with both practical applications and theoretical
extensions, enabling development of still more complex and evolving systems in business and administration.

**References**


Applications of Implementation Diagrams in System Infrastructure Modeling (Summary)

Software architecture of systems under development as well as specification of ICT equipment on which systems’ execution will be based upon are nowadays inherent elements of complete information system documentation. These aspects are in particular addressed by Unified Modeling Language (UML) by implementation-oriented diagrams. Two kinds of such diagrams could be singled out – component diagrams and deployment diagrams. The diagrams are used respectively for specifying and interconnecting software and hardware resources. The goal of the paper is to assess as well as to verify challenges and opportunities related to the applications of implementation diagrams within the development process of modern digital economy-oriented systems. After introduction to the article, the significance of the diagrams for infrastructure modeling is discussed in section 1. Section 2 is aimed at depicting functional capabilities of the diagrams. In the last section, the assessment and exemplification of the specific diagrams is
introduced as a series of mini-cases. The case studies included in the paper cover the issues of intelligent metropolitan traffic management, IS-assisted store service as well as mobile devices-based electronic ticket purchasing.

**Keywords**
Unified Modeling Language, system implementation, components, deployment, case study