Approaches to System Development Methodologies

Many paradigms for information systems (IS) development, particularly for their analysis and design, have been proposed over the years. The functional approach (also known as “process-oriented” or “traditional” approach) was very popular during the 1980s and the 1990s of the 20th century. The life cycle for developing an IS according to this approach is based on the “waterfall” model (or variations of it), which distinguishes between certain stages of development which are carried out in a serial manner, with the possibility of iterations between the
stages. According to this approach the IS is built from functions (or processes) that are connected in a complex manner, and there are constant flows of data between functions. IS analysis focuses on the identification and definition of the functions and the dataflows.

Common methodologies that support the functional approach are system structure analysis (SSA) for analysis and system structure design (SSD) for design. SSA (DeMarco, 1978; Gane & Sarson, 1979) is based on the use of data flow diagrams (DFDs), which define the functions of the system; the data stores within the system; the external entities, which are the sources of inputs and the destinations of outputs of the system; and the dataflows among the aforementioned components.

Early development methodologies such as SSA emphasized the functional aspects of system analysis, that is, functional modeling, but neglected the somewhat structural aspects, that is, data modeling. This was remedied by enhancing those methodologies with conceptual data modeling methods, usually the entity relationship (ER) model (Chen, 1976), that is used to create a diagram of the data model, which is later mapped to a relational database schema. The role of entity relationship diagrams (ERD) in data modeling can be viewed as equivalent to the role of DFD in the functional modeling. For years, DFDs and ERDs have complemented each other in the traditional development methodologies (see, for example, Hoffer, George, & Valacich, 1999; Yourdon, 1989).

SSD methodology for systems design (Yourdon & Constantine, 1979) is based on the use of structure charts (SC), which describe the division of the system to program modules as well as the hierarchy of the different modules and their interfaces. Certain techniques have been proposed to create SCs from DFDs. But the transition from DFDs to SCs is problematic because DFDs are basically a network structure, while SCs are hierarchical. Despite various guidelines and rules for conversion from one structure to the other, the problem has not been resolved by those methodologies (Coad & Yourdon, 1990).

Architectural design of information systems based on structural analysis (ADISSA) methodology (Shoval, 1988, 1991, 1998) resolved this problem. It uses hierarchical DFDs during the analysis stage (similar to other functional analysis methodologies), but the design is based on transactions. A transaction in ADISSA is defined as a process that supports a user who performs a business function and is triggered as a result of an event. Transactions will eventually become the application programs. At the beginning of the design stage the transactions are derived from DFDs (according to certain rules) and the process logic of each transaction is defined by means of structured programming techniques, for example, pseudo code. Based on the DFDs and the transactions, ADISSA provides structured techniques to design the user interface—a menus tree (Shoval, 1990), the inputs and outputs (forms and reports), the relational