

Modeling is a key activity in conceptual design and systems design, and a crucial step in the development of simulation models [6, 7, 8]. There is broad agreement that is important to involve various experts, stakeholders and users in a development cycle. If users are not involved in systems analysis tasks, their problems, solutions and ideas are difficult to communicate with the analyst. This often results in poor requirements definition, which is the leading cause for failed IT projects [9]. Thus, it is important to organize systems modeling process collaboratively between analysts, developers, stakeholders and users.

Collaborative modeling is the process of joint creation of a shared graphical representation of a system. While there is means to verbally explain models, such as metaphors, a graphical representation is often more effective. (“A picture tells more than a thousand words”). To use graphical representations as a basis for discussion, it would be useful if all the stakeholders can be actively engaged in the construction and modification of such models [10].

Collaboration means all members establish goal together and solve problems in a cooperative manner to reach common objective [14 (2)]. Collaboration is usually divided into two types: data-based collaboration and reciprocation-based collaboration. The former, such as data transmission and design technique exchange, narrates mainly the sharing of data and knowledge through the integration of artificial intelligence and database technology. While the latter discusses the situation of real-time and synchronous operations between the participants in a collaborative process, such as real-time design [15 (3)]. The collaborative software design assembles many relative personnel who simultaneously participate in the software development process, including designer, developer, project manager, manufacturer, supplier and marketer. Members from different locations can communicate and discuss issues together to concurrently carry out the software design and modification via the network that makes design results more in accordance with the consumer’s requirements [16 (7)].

Software engineering involves teamwork and communication of many kinds. Specific examples include:

- In agile processes, such as eXtreme Programming, pair programming requires very close collaboration focused on the same artifact. In collaborative software engineering (CSE), the pair members need not be spatially co-located.
- Development activities such as analysis, design, testing and coding maybe carried out by different combinations of individuals. CSE-mediated discussions are potentially a valuable way for effective communication and feedback between and within these groups.
- When correcting defects, team members may consult former team members, currently assigned to other projects, in order to determine the rationale for some design feature which has subsequently been identified as problematic.
- Refactoring often involves relatively minor but widespread changes. Users

who are kept informed of such activity are able to avoid potential contacts [19].

Various collaboration models are normally defined by time and location of collaboration and operation of collaboration such as close coupled and loosely coupled [14 (2)]; face to face collaboration, synchronous distributed collaboration, asynchronous collaboration, asynchronous distributed collaboration [17 (8)]; mutual collaboration, exclusive collaboration, dictator collaboration [18(9)]. The strength of face to face collaboration, synchronous distributed collaboration, close coupled collaboration and mutual collaboration is to provide on-line compiling and real-time operating functions for participants to reduce process duration. And its weakness is the difficulty of reviewing and auditing personal performance in an effective way. On the other hand, the strength of loosely coupled collaboration, asynchronous collaboration, asynchronous distributed collaboration and exclusive collaboration is no need to gather all participants to accomplish the work with a specific location at one time. The weakness of this kind of collaboration is the work breakdown structure may cause potential impacts on task completion due to individual delay [14, 17, 18].

For modeling tasks, Unified Modeling Language (UML) or UML based languages such as SoaML, SysML, etc. are usually used in software modeling. The UML is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system [12]. There are 9 types of diagrams to describe the whole system in UML.

Real-time collaborative modeling envisages a joint creation of models from several geographical places by using several computer machines in real-time. Recent advances in software engineering allow collaborators to synchronously edit artifacts. From an engineering perspective, adding real-time, multi-user collaboration to single-user applications is a challenging task as it requires the implementation of features such as conflict resolution as well as propagation and visualization of updates in real-time [13].