ABSTRACT

With the introduction of Object-Oriented Analysis and Design (OOAD), the development of Software Engineering has given rise to enormous opportunities for a more systematic specification and construction of software systems.

Despite the advantages that object technology can provide to the software development community, the fundamental problems associated with identifying objects, their attributes, and methods remain the same. It is largely a manual process driven by heuristics that software analysts acquire mostly through experience. While a number of methods exist for requirements development and specification, very few tools help analysts in making the transition from textual descriptions to other notations for Object-Oriented Analysis (OOA) and other conceptual models.

A method for the direct phase transition of analysis to design, particularly from Use cases to Class diagrams using Object-Oriented Design (OOD) based on mental heuristics has been devised. The identification of nouns or objects based on mental heuristics, entirely different from the existing Structured Analysis and Structured Design (SASD) methodology, has been demonstrated. In Software Development Life Cycle (SDLC), normally after completing the analysis phase, the construction phase takes place. Here an attempt is made to bring the construction of the class diagrams in the analysis phase itself seamlessly.

A hybrid approach, which is a combination of Unified Modeling Language (UML) based Use case driven approach and object-oriented method based on mental heuristics, has been developed. The hybrid approach helps the analyst to construct database tables in the analysis phase itself.
This research work explores the capture of data, information and knowledge in the analysis phase of software development so as to enhance the quality of model building in the design phase, facilitating the phase transition in OOAD.

The analysis model and the design model have been discussed through adequate knowledge elicitation. Textual features of the analysis model have been analyzed using a heuristics approach for the identification of classes.

The concepts of heuristics and the limitations involved have been amply explained through suitable illustrations. To automate the transition from OOA to OOD, a new AI software tool has been developed.

The collection of Use cases logically decomposes the functionality of the system. Database tables are also used frequently to fully describe Use cases. The tables form internal actors and to distinguish them from external actors, they are designated as Agents. Iterative approach has helped to get the basic flow first and alternative and exceptional flows (or, at least some of them) later.

The author has also proposed a user interface with multicolours which enhances productivity and quality of the results in the design process. The user interface is capable of identifying the relevant nouns, classes, objects etc. in a faster way than manual construction of class diagrams.

The components of Natural Language Processing (NLP) like parsing, syntactic analysis, morphological analysis etc. were made use of in the identification of candidate classes.

The Z specification technique is used to refine the proposed methodology. To choose the apt class, the Z specification can be used. The functional decomposition is achieved through Use case diagrams, and application of Z specification technique is restricted to individual Use cases.
Finally the research could be summarized as follows:

- Analysis of the information contained in Use cases to extract candidate classes.
- Use of structured English notation to represent Use case textual descriptions.
- Preliminary data design and amendment of Use cases that relate not only to actors but also to Agents i.e. database tables. Elicitation of candidate classes was made possible by including Agents in the Use case diagram.
- Use of Z specification technique, as a refinement of the proposed methodology.
- Application of the proposed methodology to Hospital Patients Care Study.
- Closer examination of post conditions contained in Use cases to tease out the functionality needed from classes and their collaboration.

The proposed methodology could be applied in the future for real time systems with the use of Vx works or other Real Time Operating Systems (RTOS). Concurrency control between processes becomes important then. While the Activity diagram is suitable for modelling concurrency, Z specification technique may not adequately suit such specifications. The use of other formal techniques can be considered in future.