# From Software Visualization into Software Perception

Eng. Kais Samkari                                    Dr. Ammar Joukhadar[*]

## Abstract

The famous 80s Software Crisis is still growing but simply we don't notice it. The non-existential nature of Software products is the main reason behind the lack of attention to the fact that there is a continuous crisis in the software industry. To face this problem, many researchers adopted Software Visualization techniques which allow users to navigate within pictures and scenes generated from a Visualization Metaphor like cities and social relations. Actually, these images and scenes were less complex than source code files, and gave a semi comprehension of a part of the system which was visualized by manual scanning a huge volume of data. Today, Software visualization is no more sufficient especially with the increasing demand for very large enterprise systems which totally exploit the complex web-based technologies. Applying Software Visualization on these applications generates images which are complex and huge yet like source code and on the other hand don't efficiently solve the problem of comprehension. The goal of this paper is to present an alternative method which is called Software Perception in order to achieve a real solution towards intuitive and task oriented software comprehension. That's done by changing the software granularity from the low level source code into components that are semantically saturated, in comparison to the transition made earlier from the Bitmap images to Vector images. The adoption of the proposed approach has direct implication on reducing the size of the data to be handled, and allows direct access, instead of roaming and manual navigation, to results of specific and essential tasks from different stages of Software Engineering classified by known roles such as Reverse Engineering for System Analyst, Performance Tuning for System Designer and Debugging for Software Tester. Results of using Software Perception techniques for a re-engineering task of a relatively large enterprise system showed 91% reduction in time and cost regarding the execution of the task.