Title: A comparative study to evaluate software development methods with respect to empirical modelling principles.

Weighting: 50% for both the model and the paper.

Abstract:

When developing software, it is important to consider what methods will be followed. This is something that becomes a common theme throughout the project and is important to choose correctly as it may restrict and/or complicate options that are available at a later date.

Empirical modelling presents a new method for developing software through experimentation and experience, something that can be more difficult to achieve with more traditional methods. This paper shall present a comparison of three development methods: EDEN, multi agent using abstract definitive machine (ADM) and finally object orientation. It shall examine these approaches to compare the advantages of each and the applications of empirical modelling within software development.

The idea behind EDEN is a single agent definitive language that combines some aspects of procedural programming. While it allows for new definitions to be added live, it could be considered as limited in terms of its capabilities to describe things that share the same properties.

Abstract Definitive Machine (ADM) can be used to model the notion of states by building entities who have a list of their own dependencies and actions that may be activated under certain conditions. By building on top of EDEN, it provides a more structured way of describing concepts within a model that share properties.

Object orientation has become common place for developing large scale software projects that rely on more formal semantics. This allows some partial verification of how software may behave and provides restrictions on when operations can be performed as well as what each part of the system can see. This is one thing that is not currently supported by EDEN tools and the paper shall discuss their differences and when they are most applicable.

All three methods above are to be discussed in more depth in the paper itself, highlighting the advantages and disadvantages that each has and how they could help to shape the future of empirical modelling. The paper will conclude by discussing three implementations of a model using these methods.

Description of Study:

The aims of this project are to compare and evaluate the development of models for an educational purpose using a selection of different programming paradigms. The purpose will be to link them back to empirical modelling principles and examine the possible directions that could be taken in the future.

The model of a turing machine shall be used to show the relationships between a classic computational model and how the idea of dependencies could be incorporated. This model aims to enable people to better understand the concepts of these machine and how the states and a tape can be used to model complex calculations.

To demonstrate some of the differences found in the comparison, the model shall be implemented in the three methods discussed in the paper. The advantage of constructing a model of this type is that it relies on a more dynamic way of thinking about and creating the variable number of states and links required. Each of these states has a number of properties, all of which have their own dependencies that may or may not visible and could manipulable by outside sources.

The traditional EDEN model shall be built in JS-EDEN allowing the model to take advantage of newer HTML features that could improve the ease of learning and interacting with the model. Any new features will need to be wrapped within the EM style of programming such as the implementation of buttons to ensure it maintains full use of dependencies and does not become declarative programming.

The next of these approaches is to use ADM which models the notion of state. This idea relates closely to turing machine states, allowing them to be modelled as entities who have their own dependencies. Depending on the outcome of the ADM project of another student, the turing machine model may be partially implemented using this to compare it to the other methods.

For the final paradigm, object oriented programming, an extended version of Java shall be presented that supports basic dependencies and can be translated back to standard Java. The model shall be implemented using this to examine if object oriented, typed based languages would be a direction that could be explored in the future for large scale software development while maintaining empirical modelling principles.

References:

Allan Wong *Before and Beyond Systems: An Empirical Modelling Approach*. PhD thesis, Department of Computer Science, University of Warwick, UK (January 2003).

Mike Slade *Definitive Parallel Programming - Chapter 2: Definitive programming and the abstract definitive machine*. MSc thesis, Department of Computer Science, University of Warwick, UK (April 1990).

Meurig Beynon *Reflections on Turing's approach to modelling states of mind*, Department of Computer Science, University of Warwick, UK (n.d.).