ASMs are

> Technically ASMs can be defined as a natural generalization of Finite4 The ASM Method State Machines (FSMs) by extending FSM-states to Tarski structures. Tarski structures, also called first-order or simply mathematical structures, represent truly abstract data types. Therefore, extending the special domains of FSM-computations to these structures turns finite state machines into abstract state machines [. . .].

Basically there are three stumbling blocks for a designer of software and/or hardware engineers, when they want to develope a system: - size, - complexity - and trustworthiness [@boerger:2003, p.1].

The ASM-method is suited for procedural single-agent and for asynchronous multiple-agent distributed systems.

The method bridges the gap between a human understannding and formulation of real-world problems and the deployment of their algorithmic solutions (implementation as a software and/or hardware machine).

The ASM-method enables engineers to - develop a *ground model*. The ground model represents a correct and complete human-centric task formulation. This is the result of the requirements captuure process. It is basically the binding contract between the application domain expert (usually the customer) and the system designer - refine the ground model. Several intermediate models constitute a *hierarchy of refined models*. - link the most detailed specification to generated code which should show that the model of the system is correctly solved according to the ground model (the contract with the customer)

The ASM-method lets the developer/engineer decide at any given point, which layer of abstraction he chooses. As any machine can have a function which is more or less powerful (which resembles the granularity of abstraction).

The most important practical benefit of the ASM-method is to provide a simple and precise framework to cimmunicate and docment design ideas and a support for an accurate and checkable overall understanding of complex systems.

ASM-method makes heavy use of pseudo-code. But what does it add to the long existing method of pseudo-code development? To read and write ASMs no knowledge of the underlying theory is needed. Though it is the mathematical underpinning which makes the method work. The ASM-method “complete(s) the longstanding structural programming endeavour by lifting it from particular machine or programming notation to truly abstract programming on arbitrary structures”.

ASM Tutorial

ASM also can be understood as framework which integrates the following activities and techniques:

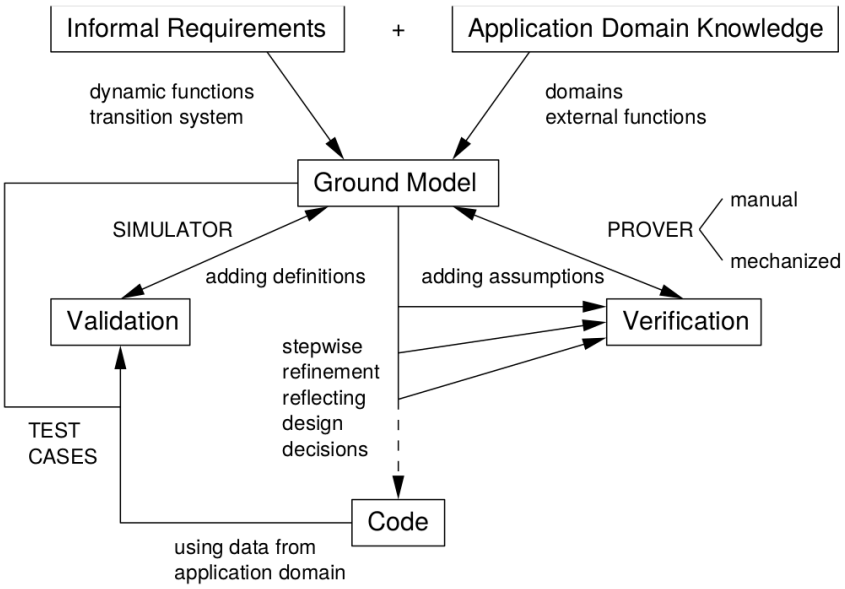

Figure 1: ASM-based methods and models

ASMs are extended Finite State Machines (FSM)

This is an inline math argument $30000 = x$ this is again normal text