Today’s and, even more so, the future development of embedded systems faces a variety of challenges. Key success factors to meeting these challenges are suitable concepts for abstraction and structure at different levels of granularity. The result of these concepts is a seamless development approach that heavily facilitates reuse and automation. A basic requirement for such a seamless approach is a clear notion of a system that is formalized by a comprehensive modeling theory. According to this modeling theory, a modeling framework has to provide appropriate models and description techniques for modeling the different aspects and artifacts of system development. This section explains these conclusions and introduces the idea of system and the modeling framework. It also references the modeling theories used in SPES.
3.1 Motivation for the SPES Modeling Framework

The aim of model-based development is to use models as main development artifacts in all phases of the development process. It promises to increase the productivity and the quality of the software development process by raising the level of abstraction at which the development is done, as well as the degree of automation, with the help of models that are tailored and appropriate for specific development tasks.

Even though adopted in practical development of embedded systems today, model-based development approaches often fail due to the lack of sufficiently powerful modeling theories and missing integration of theories, methods, and tools. The models applied in the development process are based on separate and unrelated modeling theories (if foundations are given at all), which makes the transition from one model to another unclear and error-prone.

3.2 Characteristics of Software-Intensive Embedded Systems

An embedded system can be characterized as a technical system that operates in a physical and technical environment and is built by means of technical resources that collaborate in order to achieve an overall purpose (see [Braun et al. 2010]). Embedded systems monitor and control their environment using variables that refer to specific properties of the environment (e.g., physical or technical properties; see [Parnas and Madey 1995]). IEEE Standard 1362 states that a system can be characterized as “software-intensive” if the software of the system is the major technical challenge and perhaps the major factor that affects its schedule, cost, and risk (see [IEEE 1362]). Typically, software-intensive embedded systems consist of software and hardware.

Software-intensive embedded systems are widespread in our daily life. They can be found in many application domains such as automation, healthcare, consumer electronics, avionics, transportation, and automotive.

Software-intensive embedded systems exhibit some characteristics that have a far-reaching impact on the corresponding engineering and modeling approach with which they are developed: