MODELING COMPLEX BEHAVIOR SIMPLY WITH EMBEDDED SYSTEM ENGINEERING

Vesa Salminen
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Balan Pillai
Real Time Systems Inc.
Fairfax, VA 022030, USA

Abstract

Newton's second law as written appears simple, but how to use the law perceptually is a complex task. Most of the cases it is good for conceptual discussion or development. Normally the sign convention is not shown and has problems to solve. Due to this fact, many do not fully understand Newton's second law. Modeling complex behavior is like the Newtonian law; people tend to misuse. In this article, it is experienced modeling the complex behavior simply with embedded system engineering scheme, which is conceptually a new approach. In cases where applied Newton's second law for the center of gravity G but the correct point in the formulation is the center of mass C. This is why the subscript c, not G is chosen. Nevertheless, in many engineering applications, the gravity field may be considered uniform, hence the center of mass and the center of gravity coincide, C = G, even though they are different by definition.

Keywords:
Complex, embedded, Newton's second law, complex behavior, system engineering, Internet Protocol, chip-on-system.

0 INTRODUCTION

"Thing should be made as simple as possible, but simpler," – said, Albert Einstein. Size does not matter. When it comes to modeling complex behavior, it can be too big to do the job right. As real-time systems designers, we need to specify – construct, visualize and document the dynamic behavior of a complex system. It is expressive enough to define the problem at hand, and do so economically. A Turning Machine is mathematically powerful, and the binary number system is economic in its use of symbols, but neither is expressive enough to be useful nor modeling. When one model a complex system, one has to know it is right. Integrating the subsystem software that is almost right is wrong, and wrong software can kill, mains, or incur significant economic losses. It is true that when market forces to get it right quickly but the less investment required, the better. The idea is the model build to communicate. No matter how correct or precise the modeling tools are, and no matter how stunning our own mastery of them, others must also be able to understand the models. We model systems today, whose complexity represents an exponential increase over previous models.

1 THE MODELING CONCEPT

The modeling complex behavior simply provides a rich set of modeling tools to model behavior, which we may divide into groups. The heart of the tool set is the routing-chart, a hierarchical state machine that may be associated with a class or subsystem (neuron or sigmoid functions). The route-chart also allows the developer to add dynamic actions to routes or transition-nods. We can incorporate a second-set of tools models collaborations of various kinds, such as net-synchronous of various kinds, asynchronous communication between classes (or modules) or say, instances. We can also use essentially the same flow-scheme showing particular scenarios one by one in layers.

The semantic guide will specify the routing-chart. It will inform the number of elements off-hands. Understand in detail what they are; know what they mean when executing and finally it will indicate how they relate to one another syntactically and semantically. Will determine of its use today and want to train.

2 APPLYING THE MODELING CONCEPT

One popular mechanism to manage and apply complex system is to use hierarchy to zoom in from a larger system to component pieces or zoom out to hide details. In a well-formed hierarchy, the complex system is transparent. It simply groups, state machines together with no additional semantics. In a nontransparent hierarchy if employed, the super-states have meaning, incorporating interactions with initial state, final state, deep and shallow histories. Therefore, the meaning of a dynamic model is dependent on its context that is akin to saying that a change in the outline changes the meaning of a book. To illustrate, consider an automobile with cruise control system that can be turned on or off any time. It is generally a simplified but complex system to model. One way to model this is to decompose or re-engineer the model of automobile into various subsystems or cluster them including the cruise control. The layered hierarchical setup facilitates the route-chart using the driver when turns cruise control off, one exists the super-state that represents cruise control. This complexity is simplified in figure 1 below.

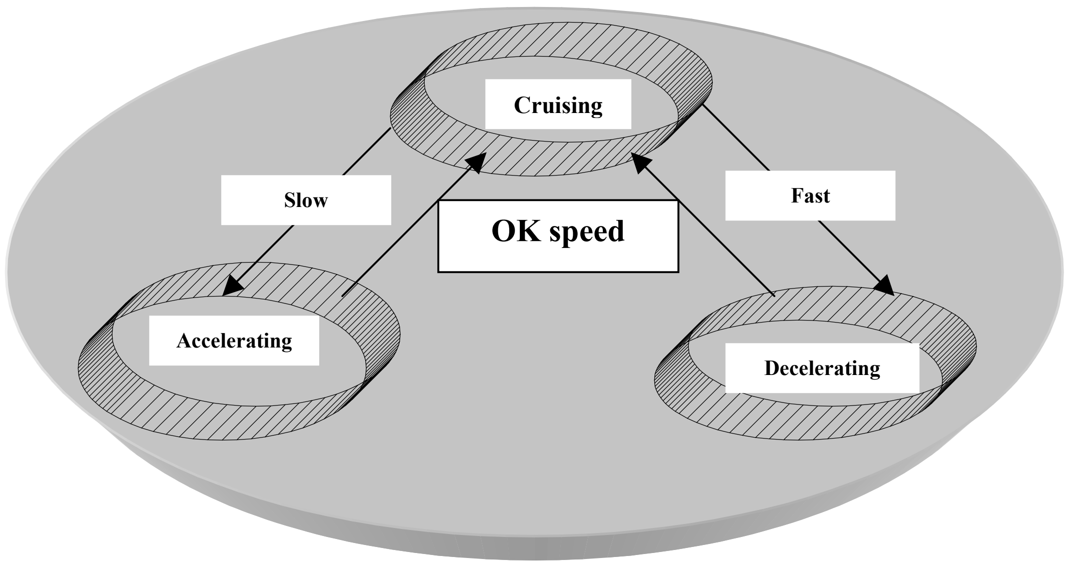

Figure 1: A modeling concept of complex behavior simply.

The other, transparent, way is to model the button that turns the system “on and off”, and communicate between that and the other things in the complex system as peers, instead of netting them. This requires no additional constructs. No new constructs, just a zooming out from a single state machine to a group of flat, related, peer state machines; complex behavior systems.

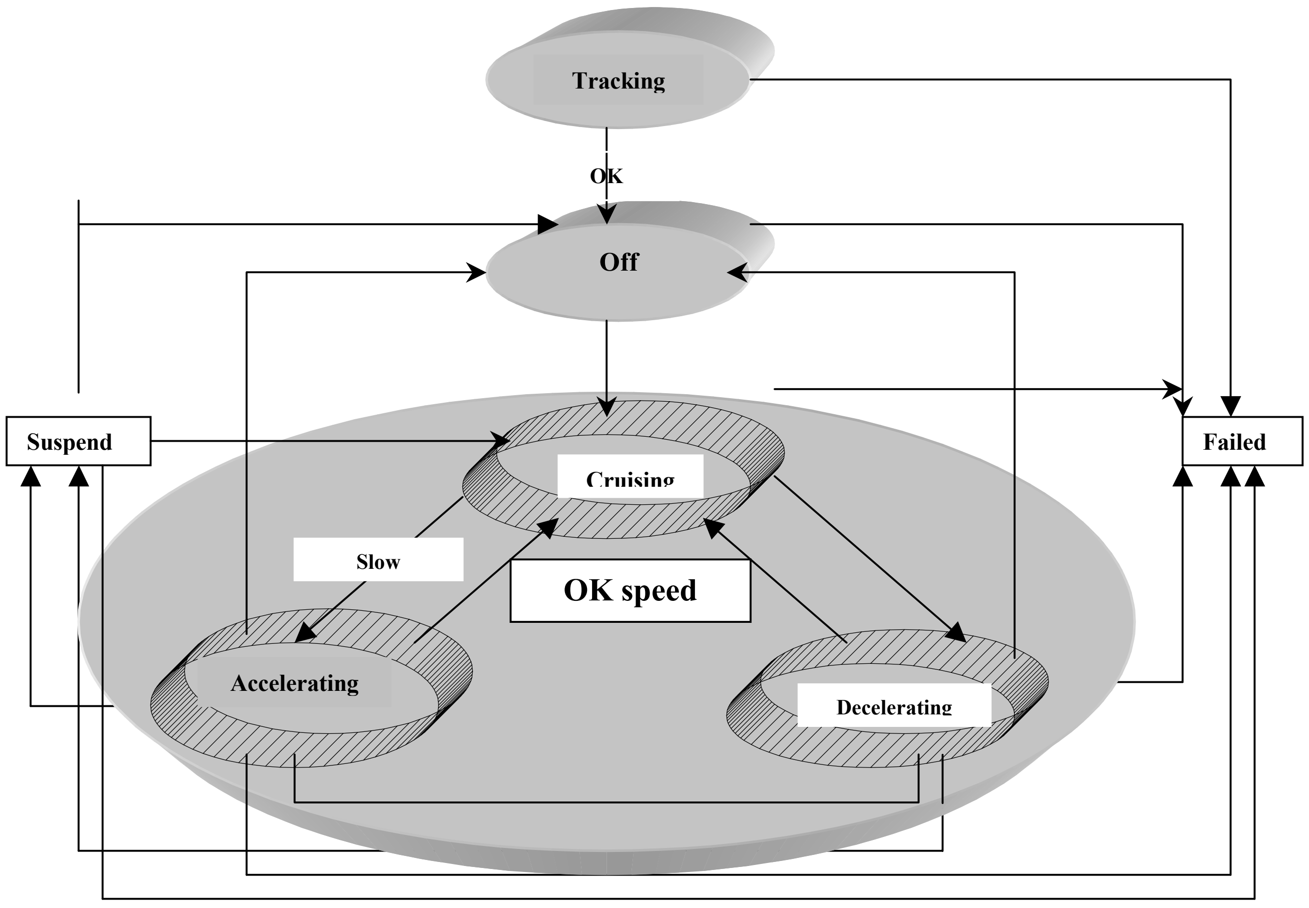

Figure 2: Implicates the complex system that deals additional requirements.

A complex system that deals with additional requirements, then we have a reasonable attempt to draw at figure 2 above. This shows that the complexity is increasing due to other action dynamic properties.

It is worth mentioning at this stage that the act of partitioning a model in this manner, as simply as possible or better still, starting from bottom and working up – almost always exposes problems that were not visible in hierarchical state mechanisms. Another example is added to draw out a situation when one tries to model an atom. The atom may be considered to consist of a central, relatively heavy, positively charged core surrounded by the proper number of electrons to produce a neutral charge on the whole structure. The electrons act as if they were particles having a definite mass and obeying Newton’s laws of motion and Coulomb’s law of attraction of oppositely charged bodies. The basic simple model of the hydrogen atom consists of a nucleus with one positive charge and a single orbital electron with a negative charge. We add complexity, as usually do. Let us assume that if the orbit of this single electron is circular, there are two equal and opposite forces acting on this particle. The first force is due to Coulomb’s law of attraction. The second one acting on the electron is due to the orbital velocity of the rotating particle. Again, the energy possessed by the rotating electron is composed of two parts – a potential energy and a kinetic energy. Now, the foregoing description of the hydrogen has been simplified to model.

3 EMBEDDED SYSTEM ENGINEERING

TESHNEHLAB & WATANABE (1999) (1) reports that the intelligent control of these kinds of models may be based on flexible neural networking system. PILLAI (1999) (2) has also shown that many schemes can be implemented. In his work it has proven that the papermaking complexity can be analyzed and reengineered the wet-end of paper producing machines. Normal way of doing this is to adapt a control strategy. Some of these technologies are changing the continuous time vs. discrete time relations. The relationship between continuous-time state space system and non-linear autoregressive with exogenous representations is simplified below in figure 3. Systems designers create one algorithmic model that is later bound to particular image types and floating or fixed-point properties. This enables complex system exploration simply by changing parameters, in contrast to traditional software approaches that require extensive editing and modification of the model to convert to fixed point.

Continuous -time | Discrete-time

State equation | **State equation**

$$\begin{tikzcd} \begin{array}{l} x=f(x)+g(x)u \\ y=h(x) \end{array} \arrow[r, "\boxed{1}"] & \begin{array}{l} x[k+1]=F(x[k],u[k] \\ y[k]=h(x[k]) \end{array} \arrow[d, "\boxed{2}"] \\ & y[k]=\Phi(y[k-1],\ldots,y[k-n],u[k-1],\ldots,u[k-n]) \end{tikzcd}$$

$\boxed{1}$: Exact or approximate discretion

$\boxed{2}$: Elimination state

Figure 3. Relationship between continuous time and discrete time elimination

Physical system modeling leads, in general, to a set of non-linear differential algebraic equations. In control systems modeling, it is usually assumed that this set of equations is, or can be reduced to a non-linear ordinary differential equations. Here the algorithm one could embed to fit the solution management. The embedded system engineering with simple modeling concept is further explained. For example the virtual component co-design environment is a key component in the system-on-a-chip (SOC) platform-based methodology pioneered to increase productivity and predictability. It enables designers to integrate virtual components representing both hardware and software (HW and SW), explore complex HW and SW tradeoffs, analyze product performance, and evaluate product architectures early in the development cycle. With these capabilities, organization strengthens its industry leadership.

4 CONCLUSIONS

In this article we establish that the complex system modeling can be done simply. To integrate them into a workable format we are now able to embed the engineering systems into web-based chip hardware. It has been more than a decade since the design community has moved up a level of abstraction from gates to the register transfer level. The functional and architectural modeling capabilities introduced here represents the next step upwards in abstraction and delivers a new front end for system-on-a-chip (SOC) design.

It also provides close integration with leading Internet Protocol creation languages and technology, such as C, C++, SDL, MatLab, behavioral HDLs, and the signal processing systems with its leading-edge libraries for wireless communications and multimedia applications.

References

1 TESAHNEHLAB, M & WATANABE K (1999): Intelligent Control Based on Flexible Neural Networks, *Kluver Academic Publication, The Netherlands, pp.6-9.*

2 PILLAI, B (1999): Adaptation of Neural Network and Fuzzy Logic for the Wet-end-control and Process Management of Paper or Board Machines – A tool making approach, *APS, publication A Finnish Academy publication, Espoo, Finland.*