

MODELING OF STRUCTURAL-DYNAMIC SYSTEMS BY UML STATECHARTS IN ANYLOGIC

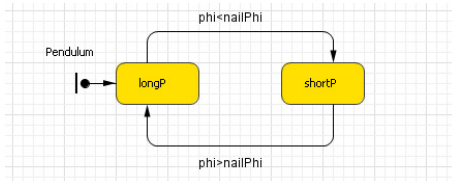
Daniel Leitner¹, Johannes Kropf¹, Yuri Karpov², Yuri Senichenkov², Yuri Kolesov²,
Günther Zauner¹

¹ TU Vienna, Institut for Analysis and Scientific Computing Wiedner Hauptstrasse 8-10,
A-1040 Vienna

² XJ Technologies Company, 49 Nepokorenykh pr. 195220 St. Petersburg
dleitner@osiris.tuwien.ac.at (Daniel Leitner)

Abstract

With the progress in modeling dynamic systems new extensions in model coupling are needed. The models in classical engineering are described by differential equations. Depending on the general condition of the system the description of the model and thereby the state space is altered. This change of system behavior can be implemented in different ways. In this work we focus on AnyLogic and its ability to switch between different sets of equations using UML statecharts. Different possibilities of the coupling of the state spaces are compared. This can be done either using a parallel model setup, a serial model setup, or a combined model setup. The analogies and discrepancies can be figured out on the basis of three classical examples.



The first is the constrained pendulum as defined in AR-
GESIM comparison C7, where the dimension of the
state space is unaltered. Second is the free pendulum
on a string, where the dimension of the state space
changes. The third example is a thermal storage model
at which between different accuracies of the discretiza-
tion is switched.

Keywords: Structural dynamics, UML, Statecharts, AnyLogic

Presenting Author's Biography

Daniel Leitner graduated in mathematics at the Technical University of Vienna. Currently he is working on his doctoral thesis about mesoscopic simulation of blood flow at the Austrian Research Centers. His research interests are numerical modeling, fluid dynamics in general, especially lattice Boltzmann methods and its application to biofluids.



1 Introduction

In this work three different structural dynamic systems are under investigation. The models are implemented in AnyLogic using its UML statecharts representation for discrete event based changes of their governing differential equations and their state spaces. In the first part AnyLogic will be described, then the different models will be presented. After that different methods of solutions are introduced with a special focus on how UML statecharts can be used to control the model structure. With this methods implementations for AnyLogic of the structural dynamic systems are given. As a conclusion the benefits and drawbacks of the hybrid simulator are examined.

AnyLogic is a multiparadigm simulator supporting Agent Based modeling as well as Discrete Event modeling, which is flowchart-based, and System Dynamics, which is a stock-and-flow kind of description. Due to its very high flexibility AnyLogic is capable of capturing arbitrary complex logic, intelligent behavior, spatial awareness and dynamically changing structures. It is possible to combine different modeling approaches which makes AnyLogic a hybrid simulator. AnyLogic is highly object oriented and based on the Java programming language. To a certain degree this ensures a compatibility and reusability of the resulting models.

The development of AnyLogic in the last years has been towards business simulation. In version 6 of AnyLogic it is possible to calculate typical problems from engineering, but there are certain restrictions. For example the integration method cannot be chosen freely and there is no state event finder.

When a model starts, the equations are assembled into the main differential equation system (DES). During the simulation, this DES is solved by one of the numerical methods built in AnyLogic. AnyLogic provides a set of numerical methods for solving ordinal differential equations (ODE), algebraic-differential equations (DAE), or algebraic equations (NAE).

AnyLogic chooses the numerical solver automatically at runtime in accordance to the behavior of the system. When solving ordinal differential equations, it starts integration with forth-order Runge-Kutta method with fixed step. Otherwise, AnyLogic plugs in another solver, the Newton method. This method changes the integration step to achieve the given accuracy.

2 Modeling

In this section three different models will be explained. They have in common that discrete events change the model structure. In the first model the state space is not altered, thus the process can be described as parameter event that occur at a discrete time point. In the second and third model this is not possible because

the dimension of the state space changes. This change of model structure and its implementation using UML statecharts in AnyLogic shall be investigated.

2.1 Constrained pendulum

A classical and simple nonlinear model in simulation techniques is the so called constrained pendulum. This model has been presented in the definition of ARGESIM comparison C7, full solutions in AnyLogic can be found in [1] or [2]. There is no exact analytical solution to this problem Therefore the results must be obtained by numerical methods. In this section a description of the model shall be given.

The motion of the pendulum is given by

$$\begin{aligned} ml\dot{\omega} &= -mg \sin(\alpha) - d\omega \\ \dot{\alpha} &= \omega \end{aligned} \quad (1)$$

Where α denotes the angle measured in counter clockwise direction from the vertical position and ω is the angular velocity. The parameter m is the mass and l is the length of the pendulum. The damping is realized with the constant d .

In the case of a constrained pendulum a pin is fixed at a certain position given by the angle α_p and the length l_p . If the pendulum is swinging it may hit the pin. In this case the pendulum swings on with the position of the pin as the point of rotation and the shortened length

$$l_s = l - l_p. \quad (2)$$

In ARGESIM comparison C7 the initial values of two experiments are predetermined:

1. The first example is given by

$$\alpha = \frac{\pi}{6}, \quad \omega = 0, \quad d = 0.2, \quad \alpha_p = -\frac{\pi}{12}. \quad (3)$$

2. The second example is given by

$$\alpha = -\frac{\pi}{6}, \quad \omega = 0, \quad d = 0.1, \quad \alpha_p = -\frac{\pi}{12}. \quad (4)$$

Both examples have the general parameters

$$\begin{aligned} m &= 1.02, \quad l = 1, \quad l_p = 0.7 \quad (l_s = 0.3), \\ g &= 9.81. \end{aligned} \quad (5)$$

2.2 Free pendulum on a string

The second example is a slightly more complicated pendulum. The massive bob of the pendulum is fixed on a string. In case of a rollover of the pendulum it

can start to fall freely until the constraints of the string apply again. This can happen if the pendulum swings higher than $\pm \frac{\pi}{2}$ and the centrifugal force is smaller than the gravitational force, see figure 1.

Thus the model has two different states: the normal pendulum movement *normal* and the free fall *fall*. The pendulum movement is given in equation 1. The equations of free fall are given by

$$\begin{aligned} \dot{v}_y &= -g \\ \dot{v}_x &= 0 \end{aligned} \quad (6)$$

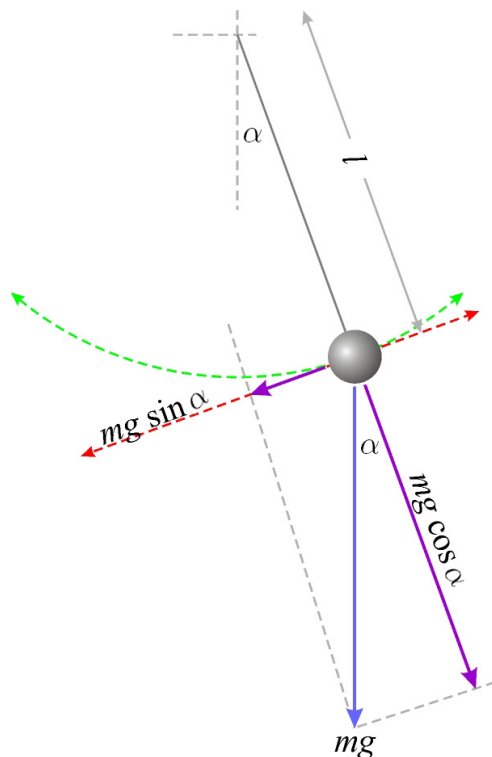


Fig. 1 Force diagram of a simple gravity pendulum

2.3 Solar heating system

The third example is motivated by the work of Nysch-Geusen, who describes a complex energy system [3]. In this work only a small subsystem will be investigated, which can demonstrate the abilities of AnyLogic to deal with structural dynamics.

A one dimensional thermal storage model shall be calculated with different accuracies that are dependent of the gradient of the temperature in the storage system. This happens for example when hot water enters the storage system. The effect is realized by using to different systems with different thermal layering as presented in figure 2.

The example demonstrates not only the dynamics but although AnyLogics ability to connect with external solvers which happens in this example because specialized finite element method (FEM) solvers are needed for the calculations.

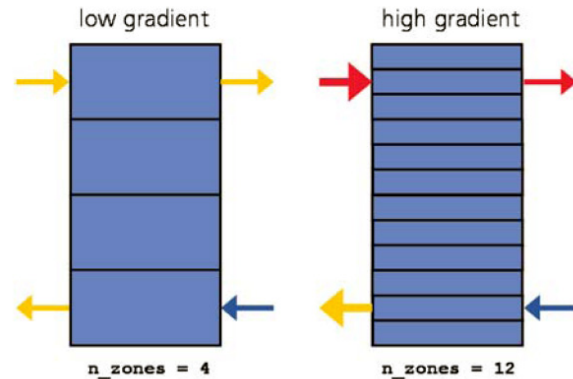


Fig. 2 Structural variable storage model, which uses a different number of zones in dependency of the current thermal layering

3 Solution approaches

New advantages in computer numerics and the fast increase of computer capacity lead to necessity of new modeling and simulation techniques. In many cases of modern simulation problems state events have to be handled. There exist different categories of structural dynamic systems which should be focused on and solved. The first class of hybrid systems are the one, where the state space dimension does not change during the whole simulation time and also the system equations stay the same. Only so called parameter events occur at discrete time points. These are the more or less simplest form of state events. AnyLogic does not differ between different types of events. The implementation of parameter events and state space changes does not differ, making AnyLogic a truly hybrid simulation environment.

AnyLogic supports the usage of UML statecharts [4]. This is a very intuitive and convenient way to describe a system which contains multiple discrete states. In the combination with dynamical equations this approach enables a simple implementation of structural dynamics. The dynamic equations or parameters are dependent of the discrete state of the model. On the other hand the transitions of the states are influenced by the dynamic variables, see [5].

The different kinds of description will be explained by means of the constrained pendulum. In this case the states are normally swinging (state *long*) or swinging with shortened length around the pin (state *short*). The discrete state of the model depends on the angle α and

the pins angle α_p . The state alters the model parameters or the models set of equations, see figure 3.

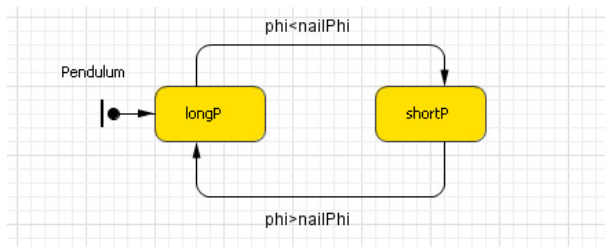


Fig. 3 UML statechart controlling the pendulum

3.1 Switching states

When the state of a system changes, often the state space of the model stays unchanged, thus the same set of differential equation can be used for different states. In this situation only certain parameters must be changed when a state is entered.

In case of the constrained pendulum the differential equation for movement stays the same for both states *long* and *short*. If the state changes the parameter length and angular velocity are updated before the calculation can continue, see figure 4.

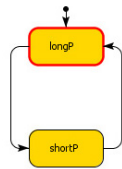
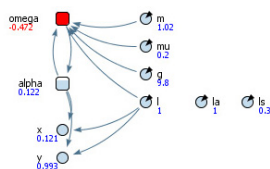


Fig. 4 The parameters of the model are changed by an UML statechart

3.2 Switching models

Often the previous approach is not possible. Sometimes situation occur where the state space of the model changes, thus a simple change of parameters is not possible. Normally the whole set of differential equations, thus the complete model, must be switched. In many simulation environments this approach can lead to complications.

In case of the constrained pendulum two differential equations are set up describing the movement of the pendulum. One describes the normal pendulum the other one the shortened pendulum. Which equation is set to be active is determined by the state diagram. When the states are switched the initial values must be passed on. The current equation must be activated and the other one must be frozen, see figure 5.

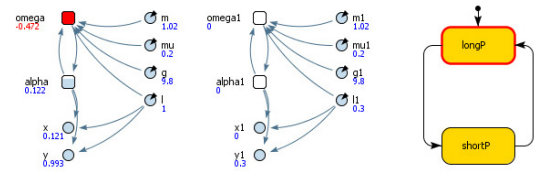


Fig. 5 The differential equations of the system are switched in dependence of the UML state diagram

4 Constrained pendulum

The implementation of the constrained pendulum has been done in two different ways. In the first approach only the parameter states have been switched corresponding to section 3.1, in the second approach the whole differential equation is switched corresponding to section 3.2. Both examples from chapter 2.1 have been simulated using both approaches. The results in AnyLogic are identical in both methods because the times of the state transitions are the same.

In the first approach the model consists of two ordinary differential equations describing the movement of the pendulum. The differential equations are realized in AnyLogic using two stock variables, the angle α and the angular velocity ω . In these equations four parameters are used length l , mass m , damping d and gravity g . Further a state diagram with states 'long' and 'short' and two transitions are used to update the equations. When the state changes length l and angular velocity ω are updated. The results calculated by AnyLogic 6 are plotted in figure 6 and figure 7.

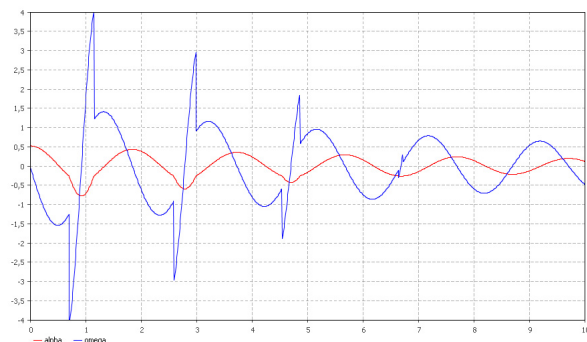


Fig. 6 The angle (red) and the angular velocity (blue) of the constrained pendulum (example 1)

The second approach uses two separate models. The implemented model consists of two times two ordinary differential equations, thus four stock variables ($\alpha_1, \alpha_2, \omega_1, \omega_2$). Both equations have four parameters separately: length l , mass m , damping d and gravity g . A state diagram is implemented analog to the first approach. If the state changes the right differential

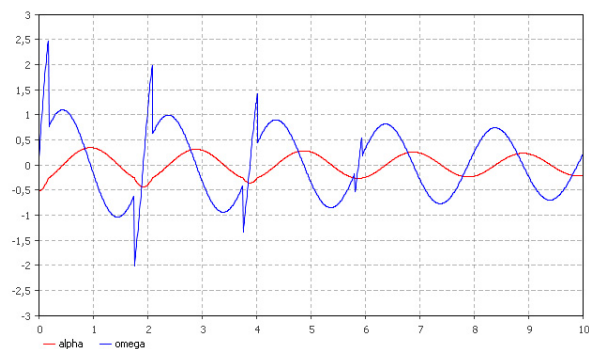


Fig. 7 The angle (red) and the angular velocity (blue) of the constrained pendulum (example 2)

equations are activated and their initial values are set, while the other differential equation is frozen.

5 Pendulum on a string

When implementing the pendulum on a string in AnyLogic, to totally different submodels must be considered:

1. The pendulum is described by the formula given by equation 1. The equation is realized in AnyLogic with the use of two stock variables, the angle α and the angular velocity ω . Further the four parameters describe the pendulum, length l , mass m , the damping factor μ and gravity g .
2. The equations of free fall uses a completely different state space. The stock variables x and y describe the position of the massive bob. In vertical direction another stock variable v_x is needed because in this direction exists acceleration due to gravity. The vertical acceleration is described with a parameter a_y . In horizontal direction a parameter v_y is sufficient.

Which model is active is controlled by an UML statechart, see figure 8. Therefore two different states are needed, the state *pendulum* for the first submodel and the state *fall* for the second submodel. Two transitions control the state of the system. The condition of the transition from state *pendulum* to state *fall* is given

$$\omega^2 r < g \quad (7)$$

expressing that the pendulum begins to fall when the gravity force is larger than the centrifugal force. The condition of the transition from state *fall* to state *pendulum* is naturally given by the constraint of the pendulum length, thus

$$x^2 + y^2 \geq l \quad (8)$$

When a state is entered all initial values for the system must be calculated from the previous submodel. When

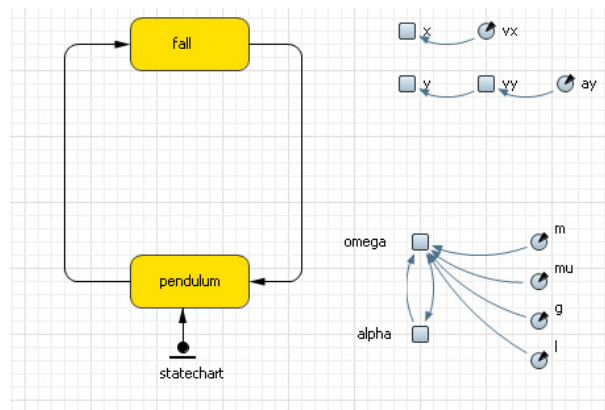


Fig. 8 UML statechart controlling the pendulum on a string

switching from *pendulum* to *fall* following initial values are preset:

$$\begin{aligned} a_y &= g \\ v_x &= \cos(\alpha)\omega l \\ v_y &= -\sin(\alpha)\omega l \\ x &= \sin(\alpha)l \\ y &= \cos(\alpha)l \end{aligned} \quad (9)$$

In the case of the transition *fall* to *pendulum* following initial values are chosen:

$$\begin{aligned} \alpha &= \arcsin\left(\frac{x}{l}\right) \\ \omega &= \cos(\alpha)v_x - \sin(\alpha)v_y \end{aligned} \quad (10)$$

The example show the possibilities of UML statecharts controlling the model structure. The main benefit of this kind of representation is the that model is clearly arranged and dependencies are shown in an intuitive way.

6 Solar heating system

The UML statechart in figure 9 controls the two different components of the problem, which solve the thermal model with a different number of nodes.

In the original solution the model is linked to external simulation software for numerical FEM calculations. In AnyLogic this link to external code can be established easily. The reason for that is that AnyLogic can be extended by arbitrary Java code. This make it possible to either start applications or to communicate with external code via the Java Native Interface (JNI).

JNI is a programming framework that allows Java code to call and be called by native applications and libraries written in other languages such as C or C++. With JNI it is possible to export and import parameters using a predetermined interface. In the case of C++ this interface look like

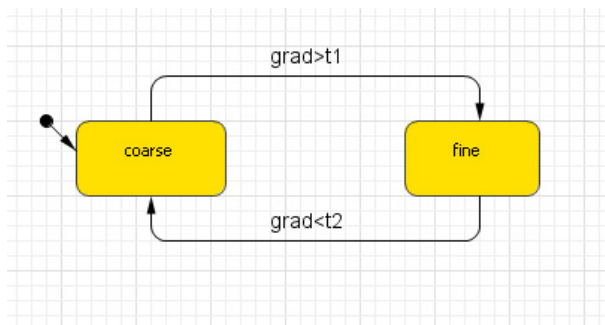


Fig. 9 UML statechart controlling the accuracy of the thermal layers

```

//C++ code
JNIEXPORT void JNICALL
Java_ClassName_MethodName
(JNIEnv *env, jobject obj,
  jstring javaStr)
{
  //For example a string is
  //imported from AnyLogic
  const char *nativeString =
  env->
  GetStringUTFChars(javaString, 0);

  //Do something with the
  // nativeString

  //don't forget to release
  // the string
  env->ReleaseStringUTFChars
    (javaString, nativeString);
}
  
```

7 Conclusion

AnyLogic is a hybrid simulator which supports a multitude of different modeling approaches, particularly UML statecharts, System Dynamics, Agent Based simulation and Dynamic Systems. In this work it has been focused on UML statecharts in combination with Dynamic Systems for the description of structural dynamic systems. In theory all different approaches can be freely combined.

AnyLogic is feasible tool to create UML statecharts and can handle structural dynamic systems in an very intuitive way. AnyLogic works strictly object oriented and translates the models to Java code and further Java code is used within the models. This ability of AnyLogic makes it easy to extend and thus applicable to a huge range of applications.

The main scope of AnyLogic version 6 is business simulation. For engineering application it is a drawback that the integration method cannot be chosen freely. Further there is no state event finder which can lead to

a significant reduction of the step size in the temporal domain.

8 References

- [1] A. Filippov and A. Kornev. C7 constrained pendulum anylogic: Hybrid modeling approach model level. *Simulation News Europe*, 32, 2001.
- [2] F. Judex. C7 constrained pendulum anylogic: Hybrid approach. *Simulation News Europe*, 35, 2002.
- [3] Ch. et al. Nytsch-Geusen. Advanced modeling and simulation techniques in mosilab - a system development case study. *Proceedings of the 5th international modelica conference*, 2006.
- [4] A. V. Borshchev, Y. B. Kolesov, and Y. B. Senichenkov. Java engine for uml based hybrid state machines. *Winter Simulation Conference*, 2000.
- [5] A. Borshchev. Anylogic 4.0: Simulating hybrid systems with extended uml-rt. *Simulation New Europe*, 31:15–16, 2001.