# Iota: An Approach to Physically-Based Modelling in Virtual Environments

Mashhuda Glencross, Toby Howard, Steve Pettifer
Advanced Interfaces Group
University of Manchester, U.K.

## Abstract

*Physically-based modelling is a powerful technique for specifying the laws and characteristics of Virtual Environments (VEs) and the objects they contain. In this paper we describe Iota, a component framework for creating and interacting with multi-body physically-based models in VEs. In conclusion, we present two case studies implemented within the framework.*

## 1. The Architecture of the Iota Framework

In this paper we describe a flexible component framework for physically based modelling in VR called Iota. This framework consists of a tightly coupled simulator engine and complexity management module together with a component for managing the VE. A high-level view of this architecture is shown in Figure 1.

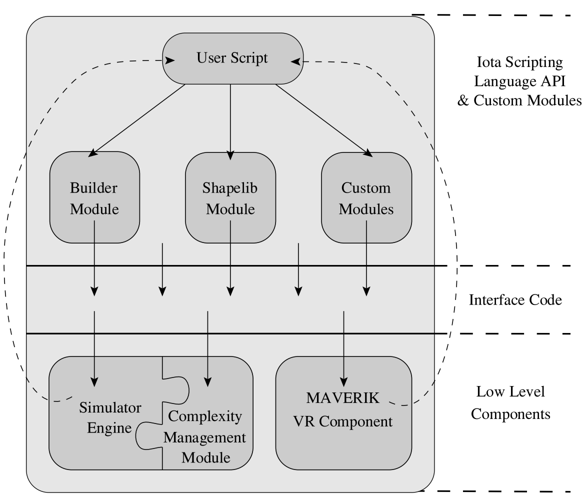


**Figure 1. The Iota framework for physically based modelling in VR**

The constraint based simulator [1, 8] is purpose-built and supports a hybrid particle/rigid body modelling approach [4]. VE functionality is provided through the GNU-MAVERIK [6] and Deva [7] systems and integrated into the framework. Above the low-level component layer lies a glue layer to enable individual components to be accessed through a single Application Programming Interface (API). On top of this layer are a collection of custom modules to support model construction, rendering customisations and any other application-specific code. User applications are implemented in a high-level scripting language (Perl [9]), and make calls to custom functionality as well as low-level functionality through the glue layer. Interfaces to all components are supported through the glue layer and this protects the user by only making available functionality which should be accessed. Low-level simulator and MAVERIK calls may make some callbacks into user code or custom modules.

## 2. Results

This section presents two case studies implemented using Iota. The first is a physically-based simulation of a Newton's Cradle, placed within an existing VE. The second is a simulation of a linked chain which can be interactively assembled and broken.

### 2.1. Case Study 1: A Newton's Cradle

In Figure 2 we show a sequence of images illustrating users interacting with a physically-based model of a Newton's cradle. The model is positioned on the surface of a desk in a VE representing our research group's laboratory. The motion of the balls in the cradle is plausibly simulated [5], and correctly maintained regardless of the number of balls raised in the cradle or the combinations in which they are raised. User trials indicated that users preferred the laboratory model VE with interactive physically-based models, rather than as a passive environment in which they could simply navigate [3].

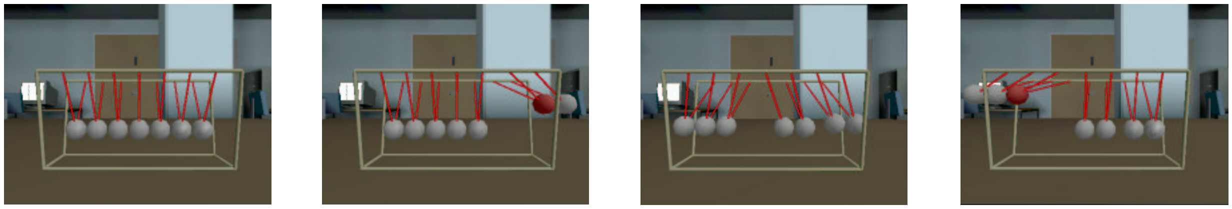

**Figure 2. Interacting with a virtual Newton's cradle**

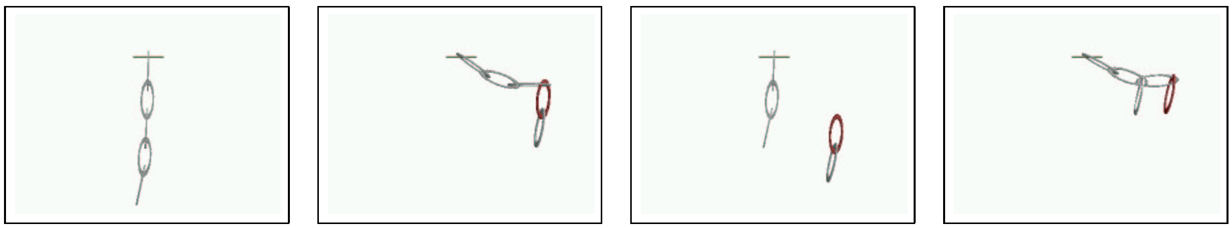

**Figure 3. Interacting with a virtual chain**

### 2.2. Case Study 2: A Virtual Chain

This case study shows a virtual chain which is allowed to swing from a point where it is suspended by a horizontal bar. The sequence of images in Figure 3 shows a distributed model of a virtual chain being interactively broken and re-assembled. In this case study, no simplifying assumptions are made in terms of the number of bodies which can occupy a joint or indeed which connections can be made or broken.

## 3. Conclusions

The Iota framework makes contributions in two areas: the simulator engine and the framework itself. Although we have used established animation techniques for physically-based simulation, we have used and augmented these techniques with functionality enabling interactive manipulation of physically-based models in a distributed VE context.

In conclusion, the Iota prototype combines some of the principles advocated by the Deva model of a hierarchy of increasingly refined environments together with a simulator engine. This is in contrast with the currently favoured-approach where the VE application may execute an application specific simulation [2] or scripted animation sequence. In particular, by implementing Iota as a component framework the flexible architecture allows us to easily customise and extend Iota, and thus make use of different solvers which may be more suited to a given application.

## References


[1] R. Barzel and A. Barr. A modelling system based on dynamic constraints. *ACM Computer Graphics*, 22(4):179–188, Aug. 1988.

[2] J. M. Cooke, M. J. Zyda, D. R. Pratt, and R. B. McGhee. NPSNET: Flight Simulation Dynamic Modeling Using Quaternions. *Presence*, 1(4):404–420, 1994.

[3] M. Glencross. *A Framework for Physically Based Modelling in Virtual Environments*. PhD thesis, Department of Computer Science, University of Manchester, 2000.

[4] M. Glencross and A. Murta. Multi-body simulation in virtual environments. In R. Zobel and D. Moeller, editors, *Simulation — Past, Present and Future. 12th European Simulation Multiconference*, pages 590–594, Manchester, England, June 1998.

[5] M. Glencross and A. Murta. Managing the complexity of physically based modelling in virtual reality. In *Fouth international conference of Computer Graphics and Artificial Intelligence*, pages 41–48, Limoges, May 2000.

[6] R. Hubbold, M. Keates, S. Gibson, A. Murta, S. Pettifer, J. C. Adrian West. Editors, and T. Howard. MAVERIK programmers guide. Technical Report MPG v5.3, University of Manchester, Jan. 2000.

[7] S. R. Pettifer. *An Operating Environment for Large Scale Virtual Reality*. PhD thesis, Department of Computer Science, 1999.

[8] J. Platt. A generalization of dynamic constraints. *CVGIP: Graphical Models and Image Processing*, 54(6):516–525, Nov. 1992.

[9] L. Wall, T. Christiansen, and R. Schwartz. *Programming Perl*. O'Reilly and Associates, second edition, 1996.