Proceedings of the TMCE 2008, April 21–25, 2008, Izmir, Turkey, Edited by I. Horváth and Z. Rusák

SIMULATING THE USE OF PRODUCTS: APPLYING THE NUCLEUS PARADIGM TO RESOURCE-INTEGRATED VIRTUAL INTERACTION MODELS

Wilhelm F. (Wilfred) van der Vegte
Faculty of Industrial Design Engineering
Delft University of Technology
the Netherlands
w.f.vandervegte@tudelft.nl

Imre Horváth
Zoltán Rusák
Faculty of Industrial Design Engineering
Delft University of Technology
the Netherlands
{i.horvath, z.rusak}@ tudelft.nl

ABSTRACT

We introduce a methodology for modelling and simulating fully virtual human-artefact systems, aiming to resolve two issues in virtual prototyping: (i) integration of distinct modelling and simulation approaches, and (ii) extending the deployability of simulations towards conceptual design. We are going to offer designers a new way of investigating the use of a product, by integrating scenarios of expected human-artefact interaction and simulations of artefact behaviour into a unified framework. Since recruitment and employment of human subjects for physical and virtual testing is problematic, we propose a fully virtual simulation method based on resource-integrated models. The models incorporate both the logical and the physical aspects of the behaviours of humans and artefacts. This paper elaborates on a pilot implementation, in particular on realizing the implementation of the physical modelling and simulation elements based on commercially available software packages. Within limitations imposed by the software we used, the applicability testing by carrying out simulations of virtual human-product interaction during the use of a product proved that human-artefact interaction could be simulated with sufficient fidelity based on resource-integrated models,. It also provided useful knowledge on the improvements needed to develop a full-fledged dedicated simulation package.

KEYWORDS

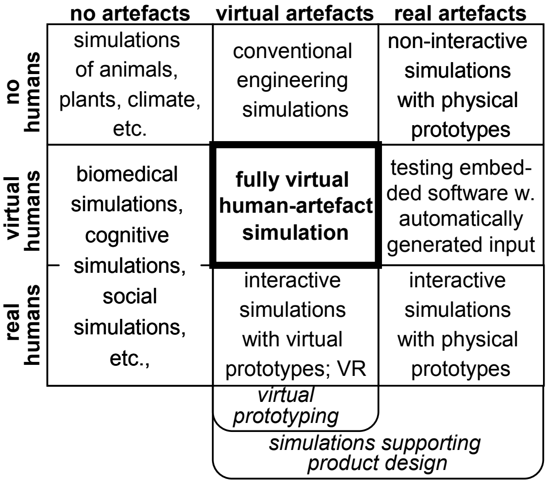
Product design, virtual prototyping, hybrid simulation, use process, nucleus-based modelling, scenarios, grasping simulation.

1. INTRODUCTION

In product design, simulations make it possible to test various aspects of the behaviour and use of products based on a prototype, i.e., a preliminary, evolving form of the product. Digital prototypes generated by a computer are in use for several decades. Such virtual prototypes are especially considered useful in conceptual design, where physical prototypes are not always available.

Computer-based engineering focuses mainly on the physical behaviour of virtual prototypes as a reaction on virtual loads, or as a result of their interaction with other artefacts. Conventionally, the engineering-oriented approaches do not consider involvement of human users. Thus, they ignore ergonomics aspects and the physical effects of interaction with humans. To resolve this omission, interactive computational simulations in which, real human subjects interact with virtual prototypes have been proposed, to make investigation of user-product interaction tangible. Actually, both artefacts and human users can be either real or virtual in a simulation (Figure 1). The second and third columns in Figure 1 arrange the different kinds of simulations based on how artefacts and humans (real or virtual) are involved. In our research, we are concentrating on the central cell: fully virtual simulation of human-artefact interaction.

The existing methods for simulating behaviour of virtual humans and virtual artefacts show a large variety in terms of the aspects considered, the types of behaviour included, and the models used. Our goal is to develop a computational solution that enables designers to jointly consider the various aspects by simulating the different kinds of behaviour together. However, we do not expect that designers will bene-



	no artefacts	virtual artefacts	real artefacts
no humans	simulations of animals, plants, climate, etc.	conventional engineering simulations	non-interactive simulations with physical prototypes
virtual humans	biomedical simulations, cognitive simulations, social simulations, etc.,	**fully virtual human-artefact simulation**	testing embedded software w. automatically generated input
real humans		interactive simulations with virtual prototypes; VR	interactive simulations with physical prototypes

Figure 1 Characterization of simulations as experiments with models.

fit from this if they have to deal with a large variety of models. The variety of models used in simulation causes a complexity that needs to be resolved in integrated simulation. Actually, this is our main research problem. Related to this problem we formulated the following research questions: ‘How can a limited set of models be developed, that can be computationally and procedurally integrated and concurrently used in simulation?’ Furthermore, ‘how can we holistically cover different types of artefact behaviour and different types of human behaviour?’

We hypothesized a solution for bringing behaviours together in a fully virtual simulation of using products. The developed solution consists of a two-layered approach, one layer addressing physical behaviour and the other one addressing logical (or ‘information-processing’) behaviour. As physical behaviour we consider manifestations of operational processes that are commonly modelled based on the laws of physics. As logical behaviour we consider (i) decision-making and muscle control by human users of the product and (ii) information-processing by the software and electronics built into artefacts. Physical behaviour is simulated by the physics simulation algorithm of a *physical simulation model*, which represents the geometry and the physical properties of humans and artefacts. Logical behaviour is simulated by processing rules that are represented in *logical models*. When the same underpinning theories, modelling concepts, data representations, relation structures, and processing techniques are used, the two models together form a *resource-integrated interaction model(RIIM)*.

The two constituents of the RIIM are *co-simulated* in order to provide a comprehensive picture of the behaviour of the human-artefact system. Co-simulation is a holistic unified simulation performed by different simulation engines at the same time. Co-simulation with a resource-integrated interaction model enables a designer to perform ‘what-if’ type of studies involving variations in the product’s design, in its physical properties, in the surroundings of use, in human users, as well as studies with different ‘scenario bundles’ of human decisions during use.

We have verified our hypothesized solution by building a proof-of-concept implementation, which was realized with commercially available software packages. It was tested by modelling and simulating the use of a conceptual product. Within certain limitations imposed by the software we used, the applicability testing proved that co-simulations could be performed as hypothesized. This strengthened our belief that, if it is further developed to a fully fledged dedicated system, our methodology of resource-integrated co-simulation can be an efficient approach for considering alternative uses and use processes in early virtual prototyping.

In the next section, a literature survey is presented to describe the current state of the art. Then, in section 3, the fundamental concepts of our solution are introduced. Section 4 describes how the proof-of-concept implementation as realized, and in section 5, the specific application to a concept product is elaborated. Section 6 discusses the simulation results and, finally, section 7 presents the conclusions and plans for further work.

2. REVIEW OF RELATED WORK AND THE STATE OF THE ART

In this section we will outline the state of the art based on the capabilities of existing simulation approaches. The objectives are (i) to obtain insight into the coverage of types of behaviours, (ii) to make an inventory of the different types of models used, and (iii) to assess the level of integration achieved by the various existing simulation approaches. Our main context is supporting designers working with 3D virtual product models. This implies three things. Firstly, we focus on achievements that are applicable in a design context (though we also include relevant achievements in other areas, such as entertainment and healthcare). Secondly, we favour simulations that can directly use 3D design models rather than simulations that require separate preparation activities. And thirdly, we focus on simulations of the behaviour of virtual models rather than methods that de-

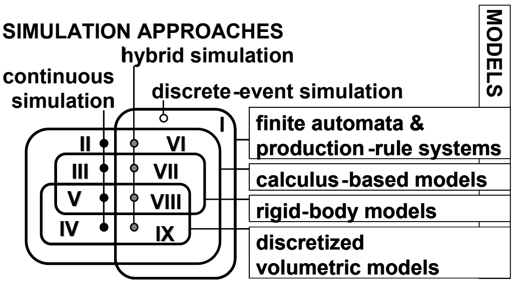


Figure 2 Characterization of simulations (left-hand side) based on the types of models used (right-hand side). The Roman numbers I-IX are referred to in the text.

pend on human subjects or physical models.

Based on the types of models used, Figure 2 gives a characterization of the basic simulation approaches that are prevailing in virtual prototyping: (i) *finite automata and production-rule systems,* (ii) *calculus-based models*, (iii) *rigid-body models*, and (iv) *discretized volumetric models*. As will be elaborated below, we found that all of the models are used for virtual prototyping of artefacts; all of the models are also used for simulating humans (but not always in virtual prototyping), and some models are used for simulating humans and artefacts together. To extend the range of behaviours covered, several *combinations of model types* have also been used in various simulations. In the Venn diagram on the left-hand side of Figure 2, not all of the overlaps that correspond to possible combinations have actually been realized. In this review we will first discuss the four basic simulation approaches, followed by the various existing combinations.

2.1. Basic simulation approaches in virtual prototyping

Finite automata and production-rule systems (I)

These logical models can be used to simulate logical behaviour. The finite automata used in such simulations can be considered as production rule systems arranged in networks. Simulation of logical behaviour is also referred to as *discrete-event simulation.* In the literature, numerous examples can be found in simulations of information processing by various virtual prototypes of artefacts (e.g., Glinz, M., 2004), in simulations for studying human cognitive behaviour in the social sciences, using simulation approaches such as ACT-R and EPIC (Anderson, J.R. et al., 2004), and in simulation of human motor control (Zeltzer, D., 1982).

Calculus-based simulation models (II)

Calculus-based models are typically used for simulation of physical behaviour, also known as *continuous simulation*. They consist of symbolic equations that specify a particular situation or a class of situations to which laws of physics apply (Bryant, C.R. et al., 2001). For our application in virtual prototyping, we are primarily interested in the two subtypes of calculus-based simulation models that can directly work with 3D models created by designers, namely, rigid-body models (III) and discretized volumetric models (IV).

Rigid-body simulation models (III)

Rigid-body models are used in conventional multibody dynamics, i.e. dynamic mechanical behaviour of interconnected rigid bodies (Von Schwerin, R., 1999). The connections between the bodies can be joints, contact relationships, and discrete compliant elements, such as springs and dampers. Conventional multibody simulation is limited to bodies in which internal deformations are absent or negligible. Multibody simulations have been applied to virtual prototyping of both artefacts (e.g., Yan, T.Y. et al., 2004) and humans (e.g., Van Deursen, D.L. et al., 2000).

Discretized volumetric simulation models (IV)

Discretized volumetric models are based on discretization in the time domain (as most computational approaches for continuous simulation),as well as on discretization of bodies in the metric space. The finite element method (FEM) is perhaps the most successful representative of this category (Zienkiewicz, O.C, Taylor, R.L., 2000). Others are the boundary element method (Brebbia, C.A., 1978), various particle-based methods (e.g., Terzopoulos, D., Fleischer, K., 1988) and so-called mesh-free methods (e.g., Tsukanov, I., Shapiro, V. 2002), to name just a few. The strength of these approaches lies in the capability of dealing with physical phenomena and effects that vary throughout 3D bodies: mechanical deformation, acoustics, heat flow, etc., including combinations of phenomena (*multiphysics*). With the exception of some particle-based approaches (e.g., Terzopoulos, D., Fleischer, K., 1988), the weakness of discretization-based simulations is that kinematics cannot straightforwardly be included in mechanical simulations. Representative examples of application of 3D-discretization-based approaches to virtual prototyping can be found in, for instance, acoustics of a virtual artefact (Tsuchiya, T. et al., 2003), mechanical deformations in a virtual human (Chow, W.W. and

Odell, E.I, 1977), and mechanical deformations in a virtual human interacting with a virtual artefact (Chu, T.-M. et al. 1995).

2.2. Combined simulation approaches in virtual prototyping

An ideal simulation approach would embrace all the behaviours covered by all the basic simulation approaches. That is, it would be capable of covering both logical behaviour *and* physical behaviour, including mechanical deformations *and* kinematics, *and* non-mechanical physical behaviours of artefacts *and* humans. Based on the findings in 2.1, the minimum of simulation approaches that should be combined to achieve this is a combination of (1) finite automata and/or production systems, and (2) discretized 3D volumetric models, in particular particle-based approaches. This corresponds to combination IX in Figure 2. Alternatively, an approach that adds rigid-body models might cover all behaviours of interest. This would make it possible to use the more conventional discretization-oriented simulation approaches such as FEM, instead of particle-based ones. Based on our findings from the literature, the actually realized combinations that use 3D models can be found in the adjacent subsets V, and VII.

Combination V: Discretization-based and rigid-body models

In mechanics, combinations of multibody simulation and discretization-based methods such as FEM have been developed. These approaches combine rigid-body simulation with deformations, but they have various limitations. For instance, 'modal analysis' is a method developed to study deformation effects of vibrations. It unidirectionally exports eigenmodes of 3D bodies from FEM to multibody simulation. Other deformations, however, cannot be investigated. Schiehlen (2006) describes an application of modal analysis to virtual artefact simulation.

Other approaches, mostly based on the method by Kane, T.R. et al. (1987) are limited to the investigation of deformations in beam-like components. Recently, Ibrahimbegovic et al. (2003) developed a solution to allow the investigation of arbitrary shapes, but it is limited to models with mostly holonomic constraints (i.e., conventional joints). For instance, bouncing cannot be simulated.

Certain commercial systems support unidirectional transfer of loads from rigid multibody simulation to FEM, to calculate stresses and deformations (e.g., Wang, S.L., 2001). Its weakness is that effects of deformations on the kinematical interaction between bodies are disregarded. This is also true for those approaches of co-simulation of *virtual humans* that subdivide the human body into rigid parts (e.g. bones), which are modelled and simulated as a multibody system and flexible parts (e.g., muscles), which are modelled and simulated as discretized bodies (Nedel, L.P., Thalmann, D., 2000). The neglection of kinematical effects of deformation makes these approaches less suitable for simulation of human-artefact interaction. In physical interactions, such as grasping, the deformed shapes of soft body parts kinematically influence the interface between human and artefact.

Combination VII: Finite automata/production-rule systems and rigid-body models

Co-simulation of logical behaviour and multibody mechanics is a combination that has actually been used for human-artefact interaction in virtual prototyping. One example is the Iowa driving simulator that was developed in the 1990s. To human subjects, this simulator presented so-called ambient traffic, which consisted of virtual pedestrians and virtual cars driven by virtual humans (Cremer et al., 1996). Physical behaviour was simulated using 3D multibody techniques, while decision-making by virtual humans was simulated with finite state machines.

Volvo Wheel Loaders applied a similar approach to virtual prototyping of a virtual wheel loader that performs various tasks in a virtual environment controlled by a virtual human driver (Filla, 2005). Human decision-making and the resulting vehicle controls were modelled and simulated with a finite state machine. The physical interaction of the loader with its environment was modelled and simulated with multibody dynamics software.

2.3. Some conclusions related to the literature study

The last two examples, in which discrete-event simulation and multibody dynamics have been combined, show that mutual interaction between virtual humans and virtual artefacts can indeed be simulated while including aspects of both logical and physical behaviour. What these approaches lack is means to simulate detailed, direct physical interaction between humans and artefacts, for instance, a virtual human hand manipulating a virtual product. The existing combinations of rigid-body and discretized-body simulation do not seem to resolve this problem either. To overcome this limitation, and to allow including multiphysics simulation, the use of particle-based models seems a good idea. If such models can be combined with discrete-event simulation, a comprehensive simulation approach for virtual human-artefact simulation would be within reach. In an optimal situation, it would be based on two types of models only. In the next section, we will underpin the fundamental concepts in order to define the resources needed for such a simulation approach.

3. SUMMARY OF THE FUNDAMENTAL CONCEPTS

In this section we will introduce the concept of 'resource integration' that reduces the need for using a large variety of models for integrated simulation of humans and artefacts. First, we present a reasoning model that explains our view on the interaction between humans and artefacts. This model identifies various types of behaviour, and subdivides humans and artefacts into functional subsystems. For these, we introduce two types of simulation models, which can be processed concurrently as a human-artefact co-simulation. The two types of models have various connections with each other and form a resource-integrated interaction model.

3.1. Reasoning model about use processes

Our hypothesis is that we can create appropriate computational subsystems by regarding the human-artefact system as a hybrid system. Within this hybrid system, we intend to maintain the distinction between logical behaviour and physical behaviour. These subsystems are described by two algorithms, which deal with only one type of behaviour. Figure 3 shows our model of reasoning about use processes based on the above hypothesis. One of the two subsets of behaviours is studied in this system as physically observable energy flows, and simulated with physical simulation algorithms. The other subset of behaviours is interpreted as information flows, and simulated by logical simulation algorithms. At the investigation of information flows, the physical phenomena that they are based on are disregarded. For instance, if voltage levels produced by a device are represented either by '0' or by '1', the output '100111' is the interpretation of a series of output voltages. The physical simulation is realized as a continuous process and the logical simulation as a discrete one.

3.2. Resource-integrated interaction model

The reasoning model in Figure 3 shows the human behaviour and the artefact behaviour as two disjunctive domains. In a real-life interaction process they are however interconnected, that is, they mutually have impact on the behaviour of each other. This relationship makes it necessary to further develop the reasoning model shown in Figure 3 into a model that expresses the integrity of the two domains. However, with a view to the large number of relationships that should be taken into consideration and to the complexity that this introduces, it seems to be unavoidable to introduce further simplifications. This is also supposed to be helpful in the realization of the functions related to the two domains as cooperating subsystems.

Actually, we introduced two major simplifications. Firstly, the behaviour of the main energy sources (i.e., human metabolic system, batteries, etc.) is not simulated. The assumption is that sufficient energy is available for the operation of other components. Sec-

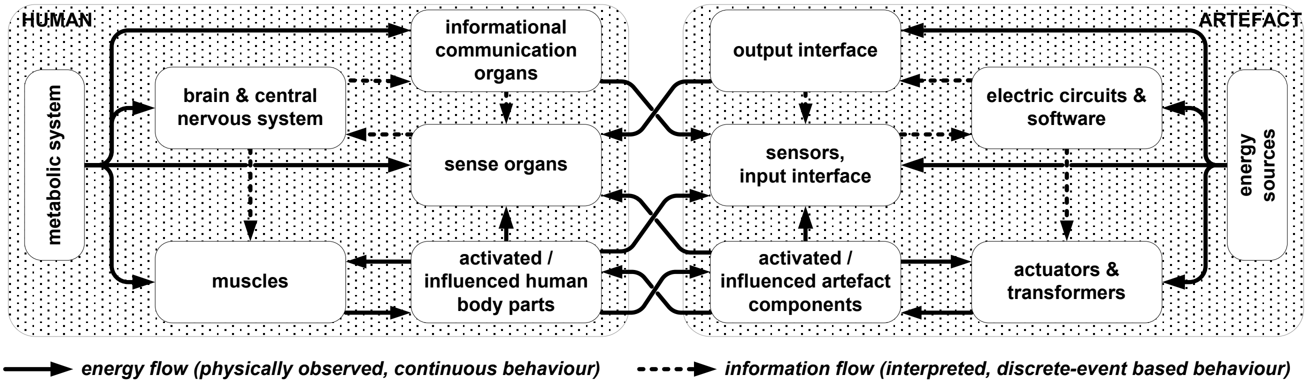


Figure 3 Reasoning model: a simplified representation of a human-artefact system and the interaction between its components. It can be considered an adaptation of the human-machine interaction diagram of (Eggert, R.J., 2005).

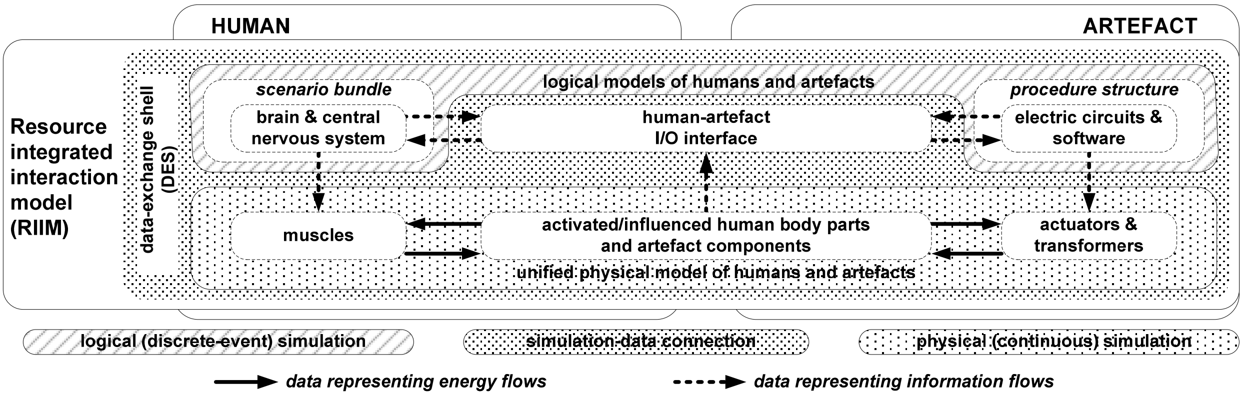


Figure 4 Simplified reasoning model for human-artefact system simulation.

ondly, information exchange between humans and artefacts is considered to be direct. It means that translation processes and the errors that they may introduce are not simulated. This reduces the functionality of communication organs, sensors, etc. to linking data streams.

Figure 4 shows how these simplifications facilitate the separation into two interconnected models: (i) a *unified physical model* of both humans and artefacts, which is used for the simulation of physical behaviour and (ii) *logical simulation models*, which are used to simulate discrete behaviour of both humans and artefacts. A *data-exchange shell* (DES) provides the connections between the two models. Both models can be implemented in different ways, which means that the DES should be adapted to them. The two models, together with their connections, which will be explained below, form the resource-integrated interaction model (RIIM).

Unified physical model of humans and artefacts

The unified physical model of humans and artefacts places virtual humans and virtual artefacts into a 3D environment, where they appear in interaction. Within the model, a distinction is maintained between (i) human muscles and artefactual actuators, and (ii) actuated and influenced human body parts and actuator parts. The components representing human muscles and artefactual actuators are controlled by the data stream from the logical model. They cause the motion of the other components of the human-artefact system by transforming control signals to energy-based workflows. These activated and influenced components, on the other hand, have velocities, distances, etc., which the system user can define as measures. These are exported to the logical simulation through the DES.

Logical models of humans and artefacts

We distinguish two forms of information processing in use processes, namely (i) human decision-making, which controls the muscle components based on what the human perceives, and (ii) information processing performed by software and digital circuits in artefacts, which controls the actuators based on what the sensors in the product measure. Consequently there must be two different types of logical simulation sub-models in the RIIM.

The simulation model representing human decision-making brings together multiple possible ways of how a product may be used by a user. Each decision can be seen and represented as making a turn in a maze of interconnected paths. In the context of user-product interaction, the route through such a network of options is commonly known as a particular scenario of use. Originating from software engineering (Carroll et al, 1994), the concept of scenarios has become widely used in various application fields of design. Since the logical model of human decision making eventually unifies a connected set of scenarios, we call it a bundle of scenarios, or a *scenario bundle*, for short. The other logical simulation model, which represents information processing by artefacts, combines all behaviours that have been programmed into the embedded software or logical circuits of an artefact, as a part of its design. We called this logical model a *procedure structure* because it represents the logical processes that the artefact is capable to perform.

Data exchange between the models

During co-simulation, the DES connects (i) the output data streams of logical models to inputs of the physical model, (ii) the output data streams of logical modes to inputs of (other) logical models, and (iii) the output data streams of the physical model, which represent values of physical variables, to inputs of logical models.

In Figure 3 and 4, the connections from each of the logical models to the physical model are direct one-way connections. Connections (ii) and (iii), however, are more complicated because they represent various interactions between the two models. This is one of the reasons why we designated a sub-module of the data exchange shell, called *human-artefact I/O interface,* in which these connections can be defined. Another reason is that, in preparation for logical, discrete-event simulations, data flows have to be converted to *events* (see 4.2.)

The values that the physical model receives from the logical models are called *control values*, and the values that the logical models receive from the unified physical model, and from each other, are called *meter values*. In a typical use process, the control values are motion instructions for human actuators (muscles) and for artefactual actuators. Meter values correspond to perceived or measured physical variables as well as to information flows produced by artefacts or humans. Based on the relations defined in the logical models, the concerned simulation algorithms react on the specified changes in the received meter values, and transfer updated control values to physical simulation.

4. PROOF-OF-CONCEPT IMPLEMENTATION

Our proof-of-concept implementation has been developed based on existing commercial software packages. This implementation has been applied to simulate the use of a concept product, a snack-dispenser. Our objective was not only to test the implementation, but also to gain further knowledge about the application potential and usability in practice.

We will first introduce the unified physical simulation model, which is the main subject of this paper. Then, the logical models will be explained briefly, and the DES that brings the two subsystems together to perform co-simulations is discussed.

4.1. Unified physical simulation model

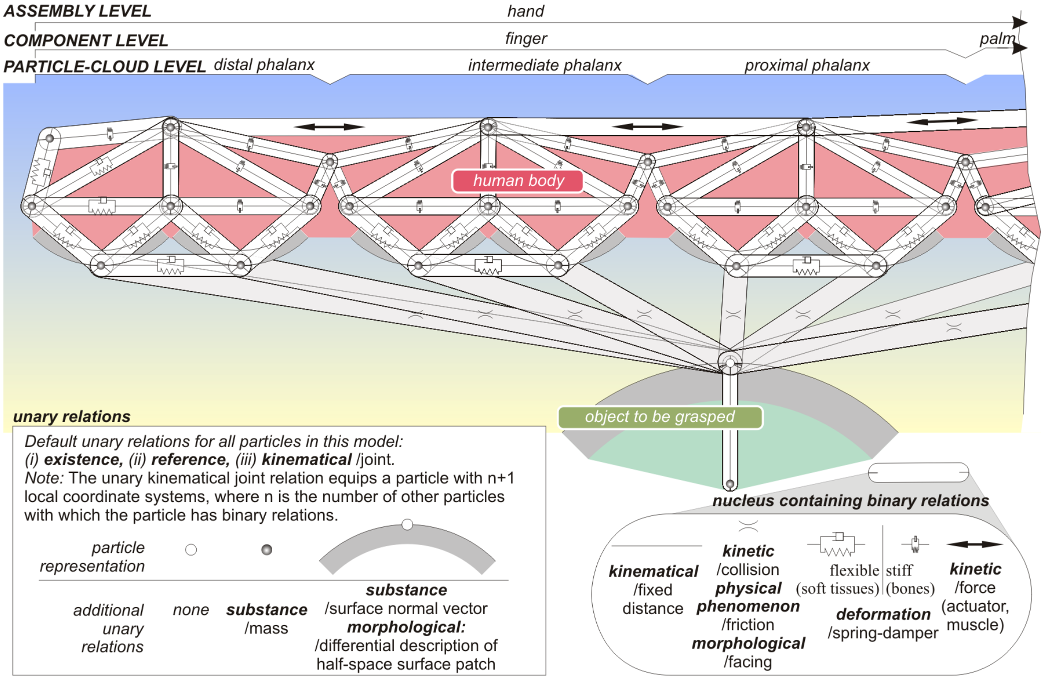
From the aspect of its implementation, a unified physical model is a 3D morphological and structural model, which contains the information needed by the physical simulation algorithms to compute the behaviour of a human-artefact system as a function of time. Thus, it includes not only geometric information, but also information about mass distribution, material properties, structure, etc.

A key issue in the development of the unified physical simulation model has been to incorporate flexible bodies in modelling and simulation. An early pilot study, in which multibody-dynamics software was used to simulate mechanical human-artefact interaction, revealed the crucial role of large tissue deformations in the modelling of physical interaction between virtual humans and virtual artefacts (Van der Vegte W.F., Rusák, Z., 2007). For instance, for adequate simulation of grasping, it is important to compute the deformations of the hand, and to take the deformation of the geometry into consideration where the hand is in contact with other entities. By doing so, we can make it possible to simulate a human hand firmly fitting around an object while grasping.

We applied *nucleus-based modelling* (Horváth, I., Van der Vegte, W.F., 2003) in order to overcome the disadvantages of the conventional volumetric and discretized physical models. Nucleus-based modelling allows, among other things, describing large deformations of flexible objects, if the constitutional equations for the given material and loading conditions are specified as relations. The nucleus-based approach discretizes an object into particles, which are connected through springs and dampers.

In a nucleus-based physical simulation model, objects are built up from *nuclei*. A nucleus is a generic modelling entity that includes one or two regions of one or two objects. These regions are interconnected by a system of relations in a particular situation. The regions and the relationships form physically coupled pairs. The idea is that all the entities in a human-artefact system can ultimately be decomposed to a purposeful composition of physically coupled pairs, and that relations, not objects, are the elementary structures to which systems can be reduced. A nucleus can capture multiple types of behaviour based on these relations. A subsystem of the human-artefact system, such as a human, a product, or a component of a human or a product, is conceived as a purposeful composition of specific instances of nuclei.

A nucleus specifies both unary relations and binary relations. A unary relation couples an object to itself, to describe reflexive properties such as mass. A binary relation couples two objects, which can be at the same level (e.g. two parts of the human body) or different levels (e.g., between a particle included in the



Default unary relations for all particles in this model: (i) **existence,** (ii) **reference,** (iii) **kinematical** /joint.
Note: The unary kinematical joint relation equips a particle with n+1 local coordinate systems, where n is the number of other particles with which the particle has binary relations.

Figure 5 Low-resolution nucleus-based model of a human finger and an object to interact with, shown as a simplified 2D representation. It is further explained in 5.1.

model of the human body and an assembly of the product). Figure 5 shows an example of a simplified nucleus-based model of a representative part of a human-artefact system: a human finger with flexible elements and a rigid simple artefact. It contains some of the most common nuclei that are discussed below.

In nucleus-based modelling, three groups of reflexive relations have been defined (i.e., existence, reference, and substance), and eight groups of binary relations (i.e., connectivity, positioning, morphological, kinematical, deformation, kinetic, physical phenomenon, and physical field). These groups are subdivided into specific relationships, for instance ‘mass’ and ‘surface normal vector’ as subclasses of the substance relation, and ‘friction’ as a subclass of physical phenomenon relations. Detail-level definitions of relations can be found in (Rusák, Z. et al., 2004).

Unfortunately, a dedicated software package for nucleus-based modelling and simulation has not yet been developed. On the other hand, in most of the commercial multibody simulation packages, various types of springs and dampers are available as standard flexible components. These packages generally also offer modelling entities that are equivalent to the relations depicted in Figure 5. This is why we decided to use a commercialized system, the multibody dynamics package of MSC, ADAMS 2007, for the proof-of-concept implementation, by creating a nucleus-based surrogate human-artefact model, and performing simulations with it. In models, to allow simulation of deformations in volumetric objects, they are first discretized into clouds of particles, which are then interconnected by springs and dampers according to the principles of nucleus-based modelling. Additionally, other relations are also defined in the physical model.

4.2. Logical models

Logical models have been defined based on the Simulink Stateflow notation, a dialect of the Statechart notation, which is a finite-automata representation introduced by Harel, D. (1987). It is available as a modelling and simulation tool within the MATLAB/Simulink environment, which makes it possible to realise co-simulations with physical simulation packages, such as ADAMS.

A statechart describes a system that is always in at least one of a finite set of states. It performs transitions between states triggered by changes in meter values. The triggers, which have no duration, are called events. An event signifies that a given meter variable received from the physical simulation crosses a given threshold value that has been specified in the human-artefact I/O interface, which sends the events to the statechart. In the statechart, instructions associated to transitions are defined. If a transition is taken, the associated instructions are carried out. Such instructions typically specify changes in the control values, which the finite automaton sends to the physical simulation model. This way, for instance, a muscle can be contracted by changing the value that has been defined as its contraction velocity from zero to a given negative number.

Since the focus of this paper is on physical modelling and simulation, we do not further elaborate on the

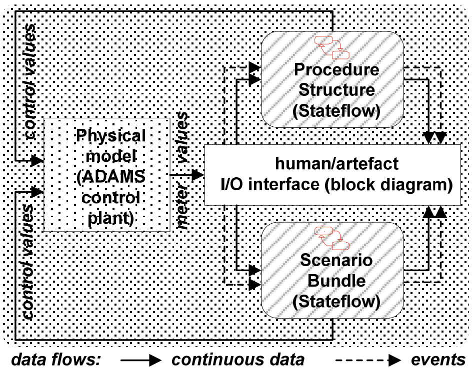


Figure 6 Connecting the physical and logical simulation models using a Simulink block diagram (simplified representation)

fundamentals of logical simulation models here. An example will be given in section 5, but for more background information the reader is referred to (Van der Vegte, W.F., Rusák, Z., 2007). We have also planned future publications on this topic.

4.3. Data-exchange shell for embedding the simulation models

In the pilot implementation, MATLAB/Simulink is used to interconnect the physical and logical simulation models. This choice was made because Simulink as an embedding environment is compatible with both the continuous ADAMS simulation package, which we used for the physical behaviour, and with the Stateflow package, which we used for logical simulation. Actually, the latter is part of Simulink

The DES is based on executable block diagrams. Figure 6 shows a schematic representation of how the reasoning model of Figure 4 has been translated to a block diagram. Within the Simulink model, ADAMS and Stateflow are running in parallel, and they are mutually exchanging values. In the next section, we will explain our method in more detail by applying the proof-of-concept implementation to an example product.

5. TESTING THE PROOF-OF-CONCEPT IMPLEMENTATION: SIMULATING THE USE OF A CONCEPTUAL PRODUCT

Figure 7 shows the human-artefact system that was simulated to test our proof-of-concept implementation. The task was conceptual design of a snack dispenser that delivers a snack after a customer has pressed a button. The picture shows the physical simulation model, which is discussed first. After that, the logical models and the connection between the models are briefly explained.

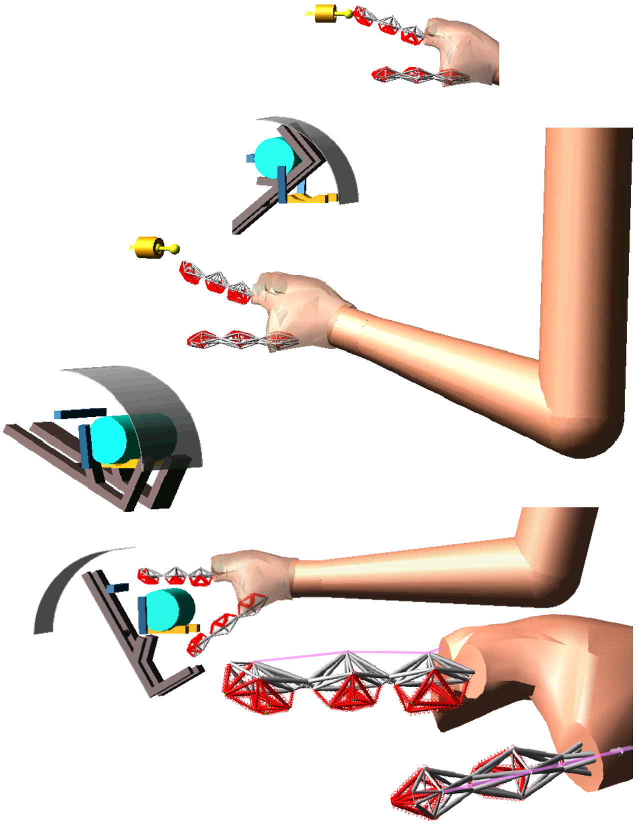


Figure 7 A snack dispenser – the human-artefact system that was modelled and simulated with MSC ADAMS and Matlab/Simulink as a case study.

5.1. Physical simulation model of the snack dispenser

The unified physical model for simulation of the mechanical behaviour of the human-artefact system in Figure 7 was created using the basic solid-modelling functionality offered by ADAMS. Figure 5 already showed how relations-based modelling with nuclei is applied to a representative part of the human-artefact system. Actually, it is a simplified 2D version of the 3D model in Figure 7, which allowed us to add explanatory graphical elements, and thus to provide insight how nuclei have been used to physically model the human-artefact system. The nuclei that have been used in the model involve the following relations: *substance* (mass of a particle), *kinematical* (fixed distance between particles), *kinetic* (collision between surfaces belonging to particles, forces between particles – i.e., muscles and actuators), *physical phenomenon* (friction between surfaces belonging to particles), *morphological* (facing of surfaces belonging to particles), and *deformation* (spring-damper connections between particles) relations. Below we will first elaborate on how the specific nucleus-based modelling entities (particles, relations, and the nuclei themselves) have been realized. After that, we will discuss the main simplifications that we applied to reduce modelling efforts and computation times.

Building nucleus-based surrogate simulation models with ADAMS

To model particles, we used the point_mass entity available in ADAMS, e.g.:

```
part create point_mass name_and_position &
   point_mass_name = .MODEL_1.particle_h &
   adams_id = 8&
   location = 0.0, -1.4E-002, 1.0E-002&
   orientation = 0.0d, 0.0d, 0.0d
```

However, point masses in ADAMS have no orientation. Therefore, we had to consider surface particles, for which the orientation can be specified because they have a surface patch attached, in a different way. Issues related to surface particles and the work-arounds we applied will be elaborated in section 6.

To model nuclei, we could use relations that were available as standard building blocks in ADAMS. For instance, a (linear) spring-damper between reference point *i* on particle *b* and point *j* on particle *d*, both of which belong to a bone, with spring stiffness *bone_stiffness* and damping coefficient *bone_damping,* has been defined as follows:

```
ude create instance & instance_name =
   .MODEL_1.spring_bone_bd & definition_name =
   .MDI.Forces.spring & location = 0.0, 0.0, 0.0
   & orientation = 0.0, 0.0, 0.0
variable modify & variable_name =
   .MODEL_1.spring_bone_bd.i_marker & ob-
   ject_value = (.MODEL_1.particle_b.MARKER_34)!
variable modify & variable_name =
   .MODEL_1.spring_bone_bd.j_marker & ob-
   ject_value = (.MODEL_1.particle_d.MARKER_35)!
variable modify & variable_name =
   .MODEL_1.spring_bone_bd.stiffness_mode &
   string_value = "linear"!
variable modify & variable_name =
   .MODEL_1.spring_bone_bd.stiffness_coefficient
   & real_value = (.MODEL_1.bone_stiffness)!
variable modify & variable_name =
   .MODEL_1.spring_bone_bd.damping_mode &
   string_value = "linear"!
variable modify & variable_name =
   .MODEL_1.spring_bone_bd.damping_coefficient &
   real_value = (.MODEL_1.bone_damping)!
```

A contact relation, which is a combined collision-and-friction relation between a particle surface of the finger (*ELLIPSOID_1149)* and the surface of the snack (*CYLINDER_5305)* has been defined as:

```
contact create&
 contact_name = .MODEL_1.CONTACT_1 &
   adams_id = 1 & i_geometry_name =
   .MODEL_1.part_particle_c.ELLIPSOID_1149 &
   j_geometry_name = .MODEL_1.snack.CYLINDER_5305
   & stiffness = 10.0 & damping = 1.0E+008 &
   exponent = 2.2 & dmax = 1.0E-003 &
   coulomb_friction = on & mu_static = 2.0
   & mu_dynamic = 1.2
   & stiction_transition_velocity = 2.0
   & friction_transition_velocity = 5.0
```

The *kinetic relations,* which correspond to muscles and actuators, exert forces or torques between particles. (From here, we will refer to these as *forces*. With a few changes, all that is said about forces can also be applied to torques). The magnitude and orientation of these forces are based on control values received from one of the logical simulation models.

We used the standard approach that ADAMS offers to model a force between two points, namely defining a single-component force or SFORCE. If our logical models would produce control values that correspond to magnitudes of forces, we could have straightforwardly defined a so-called ADAMS state variable that corresponds to an imported force value, and refer to this value in the function that defines the force's magnitude. However, our assumption has been that it is easier for a user of the modelling and simulation system to create logical models that prescribe *velocities* rather than forces, or, as a third alternative, displacements. If a prescribed force input from logical models would be required, the amount of force needed to make a component move at a realistic speed (e.g., to lift the forearm), must be known to the user of the system beforehand. This is problematic, because the force is posture-dependent and influenced by inertia effects.

In the alternative case when prescribed sequences of positions or angles are used, the user of the system has to 'plot' the whole trajectory to the end point of a motion beforehand. In the case of velocity-based control, however, the instruction can easily be formulated as *move forward* (or backward) *with a given speed until a given position or angle* (received from the object-model simulation as a meter value) *has been reached*. Still, eventually, the prescribed motions– in the form of velocities – must be translated to forces and torques to activate muscles and rotational actuators during the forward-dynamics simulation in ADAMS. This is done by defining force relations based on feedback loops using proportional-integral controllers. In the case of a muscle force, where the velocity is an angular velocity around a joint:

$$F_{ij}(t) = K_P\{\omega_{prescribed}(t) - \omega_{actual}(t)\} + K_I \int_{t_0}^{t} \omega_{prescribed}(t) - \omega_{actual}(t)dt$$

with t the simulated time, F_{ij} the force relation between points (markers) i and j, ω_{actual} the angular velocity caused by F_{ij}, and $\omega_{prescribed}$ the angular velocity prescribed from $t = t_0$. The constants K_P and K_I

(proportional and integral gain) determine the responsiveness and stability of the controller behaviour. In the case of our application example, values have been determined by trial-and-error: $K_P = K_I = 5 \cdot 10^3$. In ADAMS, this is modelled as:

```
muscle_force_i
   = propgain * (omega_desired_i-omega_i)
   + intgain * DIF(omega_error_integral_i)
```

Simplifications applied to the physical model

Since our application example merely serves as a proof-of-ideas, we wanted to reduce the time needed for modelling, as well as the computation times for simulating its behaviour. Therefore, we applied various simplifications.

Firstly, we did not put efforts in obtaining or creating a completely realistic model that accurately describes the various physical properties and behaviours of the human body. We applied a strategy of trial-and-error, varying connection arrangements between particles, and varying coefficients of friction, damping, stiffness, etc., until we had obtained a model of which the behaviour *looked* realistic. Also, we used linear springs and dampers. Outcomes of future research should provide more accurate and realistic models.

Secondly, we simplified human anatomy in two respects. As in a real human body, muscles have been modelled as linear actuators acting between points on two body parts. However, instead of two contracting muscles acting across a joint on two sides, only one 'muscle', which is capable of both contracting and expanding, is modelled. The other anatomical simplification is that the phalanges of fingers are rotated by surrogate muscles in the fingers and not by tendons connected to muscles in the forearm.

Thirdly, we modelled only two fingers of the human hand as particle clouds, since two fingers should be just enough to simulate grasping. We considered all of the other parts in the model as non-deformable. For such parts, nucleus-based models can be created with one particle for the centre of mass and a small number of additional surface particles only to define its geometry. Finally, we used low-resolution particle clouds, i.e., only 81 particles were used to represent the two fingers in the model.

5.2. Including logical models in the co-simulation

At its highest level, the *scenario bundle,* shown in Figure 8, describes the decision-making process that starts with activating the button. Once the button has been activated and released, the virtual human moves his hand towards the snack and eventually takes it. To include human latency in the model, a state hesitate has been included. If the virtual human hesitates for too long, the procedure structure (not shown) issues a command to put the snack back into storage space to prevent it from getting too warm. Once the hand is positioned close enough to the snack, a proximity sensor in the snack dispenser sends a signal (event) within the procedure structure, which prevents the snack from being put back. If the snack has been taken away and the door has been closed (snack_gone), a retry loop is included. This loop includes another hesitation (the human user is surprised to see the snack disappearing), and the next time the human realizes that he should not hesitate so long after having pushed the button (/t_hesitate = t_hesitate*0.5). Whenever the hand can reach the snack, it is grasped (grasp) and taken (remove_snack).

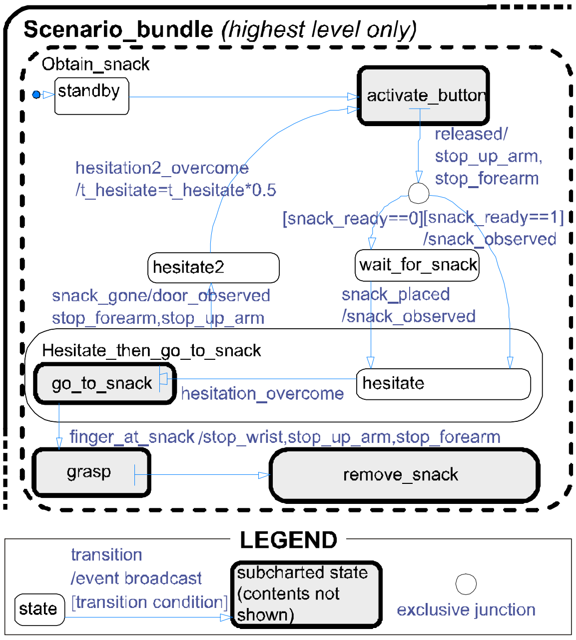


Figure 8 Scenario-bundle representation of the use of the snack dispenser

The additional functionality of delivering subsequent snacks was not included in our model, as was not the usual functionality to collect payments. The logical models have been connected to the physical model according to Figure 6. Continuous data streams containing meter values were converted to events using *hit crossing* block-diagram elements in the human-artefact IO interface block.

The co-simulation is started from Simulink. During co-simulation the logical models are animated. The continuous simulation of mechanical behaviour can also be animated during the simulation, but its 3D rendering slows down the computations so much that we chose to run ADAMS in 'batch mode'. This means that during co-simulation ADAMS runs its solver algorithm only. The 3D animation can be viewed afterwards by using ADAMS postprocessor.

6. SIMULATION RESULTS AND DISCUSSION

In a previous pilot implementation (Van der Vegte W.F. and Rusak, Z., 2007), we could successfully run hybrid co-simulations of the use of a simple product (a garbage bin), but (i) the rigid-body models we used did not allow us to simulate human grasping with sufficient fidelity, (ii) due to the simplicity of the models simulation of a wide variety of use scenarios was not possible, and (iii) the simulated product could not perform information processing, thus the concept of procedure structures was not tested. Therefore, in running simulations with the current set-up, our main objectives were to add more realism both to the physical aspects and to the logical aspects of simulation. Regarding the physical aspects the objective was to show that it was possible to simulate human grasping and large deformations by using a nucleus-based physical simulation model. Regarding the logical aspects, the goal was to simulate a variety of paths through the logical models, and to include artefactual information processing. Also, we wanted to gain a first impression of the time needed to run a simulation of a realistic use process. Our findings are discussed in the following three subsections.

6.1. Physical aspects: grasping and large deformations

The simulation of grasping (actually, the entire co-simulation) that we performed is based on simplified models with simplified behavioural characteristics. Leading to simpler models, this reduced the computation time and the number of variables that we had to include in our control loop. On the other hand, it is more difficult to model grasping with only two fingers, because the grasped object was not held as stable as it happens when it is enclosed in a full hand. Yet, after some trial-and-error we managed to successfully simulate grasping of the virtual snack by the virtual hand and holding it while carrying it away over at least 0.2m. In the animations of the physical behaviour, large deformations were clearly visible. Since our attempts to apply stiffer spring-dampers between the finger particles were unsuccessful (i.e., the snack notoriously fell out), it appears that simulation of grasping indeed benefits from including large deformations.

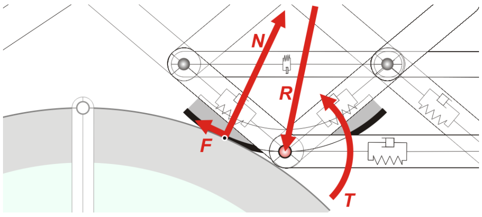


Figure 9 Contact friction force F, contact normal force N, reaction force R, and reaction torque T acting on a boundary particle (*cf.* Figure 5; collision nucleus omitted for clarity)

An issue in the creation of low-resolution particle models is choosing the size of surface patches attached to boundary particles. It is not desired that surface elements of interacting bodies pass through the gaps between each other's boundary particles. To prevent this from happening, surface patches must be sufficiently large to bridge the gaps. However, as shown in Figure 9, a contact force near the edges of a surface patch causes a reaction torque around the boundary particle. This torque must be compensated by spring-dampers to which the particle is attached. Otherwise, the surface patch will rotate around the boundary particle.

This does not reflect realistic behaviour. In real life, flexible bodies are capable of producing reaction forces at any location on their surface. In our finger model, we worked around this issue by attaching only small spherical surface patches (0.5mm radius) to each boundary particle, each with the reference point of the particle at its centre. However, this is not according to the conventions of nucleus-based modelling, which require that the boundary particle itself must be a point on the surface. Nevertheless, the workaround proved to be successful when the object to be grasped was sufficiently large (like the cylindrical snack we used in the model) and had no surface details that could have penetrated the gaps between the surface patches on the finger.

For the same reasons, it was not possible to simulate contacts between *two* discretized flexible bodies (for instance, a hand grasping a liquid-detergent bottle). This is because in our surrogate model, mutual penetration of boundary particles cannot be avoided.

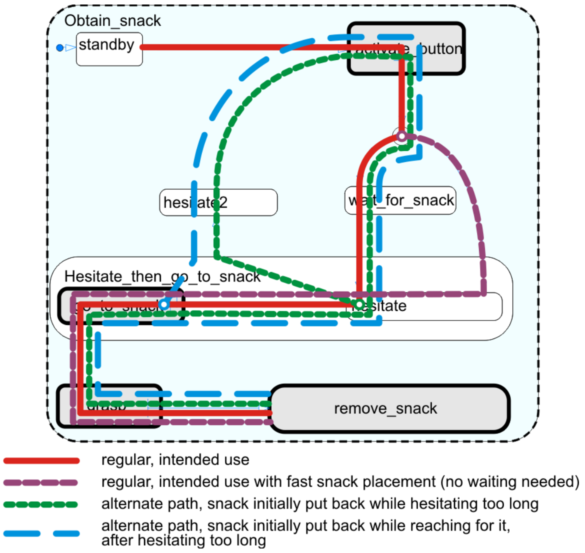


Figure 10 Simulated variety of paths through the scenario bundle (*cf. Figure 8*)

6.2. Simulating multiple scenarios based on one bundle

Figure 10 shows four of the different paths through the scenario bundle, which we were operationalized at 'playing' with the preset values of human latency (hesitation time), and with the angular velocities imposed on actuators in the snack dispenser. Other scenarios with multiple retries could also be simulated (not in the figure). The four paths in Figure 10 correspond to three different paths in the procedure structure.

6.3. Computation time

The simulation time needed for a use process with duration of 5.4s, corresponding to the fourth path in Figure 10, was 36:53 minutes on a PC with Intel Core 2 Quad 2.66 GHz CPU running Windows XP SP2, using three threads for the ADAMS 2007 simulation. During the simulation, ADAMS used 20-30% of CPU time, while Simulink used 3-5%. On average, this simulation took almost 7 minutes of computation for each second of simulated behaviour. From this we can conclude that real-time simulation is not feasible with the current proof-of-concept implementation. In a non-interactive simulation like the presented one, real-time performance is typically of minor importance because the simulated system does not need to be continuously synchronized with a real system. However, real-life application cases will likely require physical simulations with more particles, and thus more efficient algorithms will be needed.

7. CONCLUSIONS AND FUTURE WORK

In this paper the concept of resource-integrated interaction models and hybrid co-simulation of product use have been elaborated to such a level that they could be tested in an application case study. The study showed that modelling flexible bodies based on the nucleus principle made it possible to simulate human grasping, which is an important form of interaction between humans and artefacts.

Nucleus-based modelling can be used as the basis of resource-integrated interaction modelling. This was even confirmed by the proof-of-concept implementation, which was based on commercialized systems. Although these systems are imposing limitations in terms of a fully featured implementation of nucleus-based simulation models, the concept proved to be feasible and computable. We concluded that there are four shortcomings and limitations of the current proof-of-concept implementation, which need to be addressed in future work.

Firstly, simulation of physical behaviour is too slow to (i) enable real-time simulations and show realistic 3D animations during the simulations, and to (ii) allow the use of realistic, high-resolution models of virtual artefacts and humans. Although we realize that we may not have used the full modelling potential offered by ADAMS, we believe that in order to achieve the optimal performance, a system that fully obeys all rules and details of nucleus-based modelling should be programmed from scratch.

Secondly, no adequate knowledge was built into the system to enable realistic modelling and simulation of human logical behaviour (decision-making and muscle control) and of deformation behaviour of flexible objects, including human tissue. Further work is needed to incorporate what has already been found by others and to fill in the possible gaps.

Thirdly, the software and the interfaces used to prepare and connect the simulation models have not been developed from the aspect of being used by product designers. In the physical model discussed in the paper, particles and relations have been instantiated one by one, in a manual fashion. A dedicated modelling tool should provide automated conversion from CAD models, and it should also offer a library of virtual humans of various sorts. The logical models have been created using the graphical interface of Stateflow, and connections have been manually assigned by using state variables in ADAMS, and by manually creating a human-artefact I/O interface module as a Simulink block diagram. We expect that these tasks can also be supported by a certain level of automation. Additional research is needed to assess how much product designers can familiarize themselves with statecharts, or with alternative ways of building logical models.

Finally, the proof-of-concept implementation did not support simulation of all possible physical behaviours. Concerning mechanical behaviour, we need to find a way of modelling surfaces that can correctly simulate contacts (see 6.1). We expect that behavioural simulation in fields such as thermodynamics, acoustics, etc., can be also realized by using a dedicated nucleus-based modelling system.

ACKNOWLEDGEMENT

We wish to thank Chris Verheul of Sayfield International who helped us overcome the steep learning curve of the ADAMS multibody simulation software.

REFERENCES

Anderson, J.R., Bothell, D., Byrne, M.D., Lebiere, C., (2004), "An integrated theory of the mind", Psychological Review, Vol. 111, No. 4, pp. 1036-1060.

Brebbia, C.A., (1978), The boundary element method for engineers, Pentech Press, London.

Bryant, C.R., Kurfman, M.A., Stone, R.B., McAdams, D.A., (2001), "Creating equation handbooks to model design performance parameters", Procs. ICED, Glasgow, pp. 501-508.

Carroll, J.M., Mack, R.L., Robertson, S.P., Rosson, M.B., (1994), "Binding objects to scenarios of use". Int. J. Human-Computer studies, Vol. 41, pp. 243-279.

Chow, W.W., Odell, E.I., (1977), "Deformations and stresses in soft body tissues of a sitting person". J. of Biomechanical Engineering, Vol. 100, pp. 79-87.

Chu, T.-M., Reddy, N.P., Padovan, J., (1995), "Three-dimensional finite element stress analysis of the polypropylene ankle-foot orthosis: static analysis", Medical Engineering & Physics, Vol. 17, pp. 372-379.

Cremer, J., Kearney, J., Ko, H., (1996), "Simulation and scenario support for virtual environments", Computers & Graphics, Vol. 20, No. 2, pp. 199-206.

Eggert, R.J., (2005), Engineering Design, Pearson Prentice-Hall, Inc., Upper Saddle River.

Filla, R., (2005), "An event-driven operator model for dynamic simulation of construction machinery". Procs. Scandinavian Conference on Fluid Power, Linköping.

Glinz, M., (2004), "Systematically combining specifications of internal and external system behaviour using statecharts", Procs. ICSE Workshop on Scenarios and State Machines.

Horváth, I., Van der Vegte, W.F., (2003), "Nucleus-based product conceptualization – Principles and formalization". Procs. ICED, Stockholm.

Ibrahimbegovic, A., Taylor, R.L., Lim, H., (2003), "Non-linear dynamics of flexible multibody systems", Computers and Structures, 81, pp. 1113-1132.

Kane, T.R., Ryan, R.R., Banerjee, A.K., (1987), Dynamics of a cantilever beam attached to a moving base, J. of Guidance, Control, and Dynamics, Vol. 10, No. 2, pp. 139-151.

Nedel, L.P., Thalmann, D., (2000), "Anatomic modeling of deformable human bodies", The Visual Computer Vol. 16, pp. 306-321.

Schiehlen, W., (2006), "Computational dynamics: theory and applications of multibody systems", European J. of Mechanics A/Solids, Vol. 25, pp. 566-594.

Terzopoulos, D., Fleischer, K., (1988), "Modeling inelastic deformation: viscoelasticity, plasticity, fracture. Computer Graphics", Vol. 22, No. 4, pp. 269-278.

Tsuchiya, T., Kagawa, Y., Doi, M.and Tsuji, T., (2003), "Finite element simulation of non-linear acoustic generation in a horn loudspeaker", J. of Sound and Vibration, Vol. 266, No. 5, pp. 993-1008.

Tsukanov, I., Shapiro, V. (2002), "The architecture of SAGE - a meshfree system based on RFM", Engineering with Computers, Vol. 18, No. 4, pp. 295-311

Van Deursen, D.L., Lengsfeld, M., Snijders, C.J., Evers, J.J.M., Goossens, R.H.M., (2000), "Mechanical effects of continuous passive motion on the lumbar spine in seating", J. of Biomechanics, Vol. 33, No. 6, pp. 695-699.

Von Schwerin, R., (1999), Multibody system simulation - numerical methods, algorithms, and software, Springer Verlag, Berlin.

Wang, S.-L., (2001). "Motion simulation with working model 2D and MSC.visualNastran 4D", J. of Computing and Information Science in Engineering, Vol. 1, pp. 193-196.

Yan, T.Y., Chung, J.H., Choi, Y.S., (2004), "Design of the driving system of an automatic vacuum packer using functional virtual prototyping", Biosystems Engineering, Vol. 89, No.1, pp.37-46

Zeltzer, D., (1982), "Motor control techniques for figure animation", Computer Graphics and Applications, Vol. 2, No. 9, pp. 53 - 59

Zienkiewicz, O.C, Taylor, R.L., (2000), The finite element method, Vol. 1-3, Butterworth-Heinemann, Oxford.