HYBRID SIMULATION OF THE USE OF PRODUCTS BY CONTROLLING CONTINUOUS BEHAVIOUR WITH STATE MACHINES

W. F. Van Der Vegte

Delft University of Technology
Faculty of Industrial Design Engineering
e-mail: w.f.vandervegte@tudelft.nl

Keywords: Product design, designing for use, hybrid simulation, conceptual design

Abstract: *Currently, options for behavioural simulation of the use process of a product are limited. On the one hand, various types of behaviour of non-living objects, including the product, can be simulated using engineering simulation software. On the other hand, aspects of human motion and vision can be animated using ergonomics software tools. Performing complete-picture behavioural simulation in which the product and the human user react on each other's behaviours is not practicable. In this paper a new modelling and simulation approach is presented. The objective is to offer designers a method and a system to run 'complete-picture' simulations of product use in conceptual design. To make this possible, a hybrid approach is proposed. Behaviour that is commonly modelled based on the laws of physics is simulated as continuous behaviour, while information-processing behaviour is simulated as discrete behaviour. The objective is to perform continuous multiphysics simulation with nucleus-based object models as described in forerunning papers. This paper elaborates on modelling and simulation of the discrete behaviour of human-artefact systems and on how discrete-behaviour simulation is linked to continuous-behaviour simulation. Scenario structures are introduced to represent the different courses that use processes can take. Depending on what is available to the designer, scenario structures can be based on observations from real users or on conjecture. They are modelled using finite state machines. It is expected that the presented approach makes it possible to perform what-if studies with different types of human decision-making behaviour, and to investigate the effect of adaptations if the physical behaviour of the product and its environment turns out not to be as expected or intended.*

1. INTRODUCTION

During product design, behavioural simulation is frequently used to gain insight into the course of processes in which the product is involved. In a product's life cycle one of the key processes is the use process, in which the product is actually applied for its purpose. Behavioural simulations of use processes can provide a valuable contribution to 'designing-for-use' (or DfU) approaches. After all, simulations (behavioural or non-behavioural) have been defined as experiments performed on models [1] , which means that they make it possible to invest life-cycle processes such as use before a product is available in its final form. In industrial product design, the simulation model of a product is typically called prototype, i.e., a physical prototype, a virtual prototype or an augmented prototype. This paper focuses on behavioural simulation in the beginning of the design process, where virtual prototypes are preferred because they are easier to create than physical or augmented ones. A virtual prototype is a non-real, digital prototype modelled and visualized using a computer [2]. To gain insight into the use of products, studying the behaviour of virtual prototypes of products is not enough. In the literature the consensus is that, when investigating use processes, a larger system of three main components should be taken into account consisting of three main components: the human user, the product and the surrounding environment [3]. These components interact through mutual exchange and transformation of matter, energy and information. In this paper, the system will be referred to as the human-product-surroundings system, for short HPS system.

My assumption is that a simulation approach that takes into account the behaviour of the complete HPS system can be a valuable addition to the currently available methods and tools to support designing for use. It will allow designers to perform comprehensive investigation of both human aspects and system aspects. This paper introduces a method and system framework for HPS system simulation of use processes in conceptual product design, and, consequently, for the creation of virtual models on which such simulations can be performed.

The approach described in this paper particularly aims at resolving two knowledge-related issues in use-process simulation: (i) the integration of simulation and modelling approaches and (ii) the diversity of use processes. These are elaborated in the next two subsections. In section 2, related work is discussed. Section 3 introduces the concept of resource-integrated modelling and hybrid simulation of human-product-surroundings systems. Section 4 describes the method of implementing this concept into a workflow, of which the embedding in the design process is explained in section 5. Section 6 briefly explains the basics of nucleus-based object models to represent and simulate humans and artefacts and in section 7 the concept of the use state machine is elaborated, which is used to model and simulate information-processing behaviour. Section 8 describes the proposed system architecture for modelling and simulation, section 9 describes a first pilot implementation and finally sections 0 and 11 present the conclusions and plans for future work.

1.1. Integration issues

The majority of current systems that can be deployed to simulate use processes focus either on artefact behaviour without taking into account the role of the human user, or focus on certain ergonomics-related human aspects without considering artefact behaviour. Artefact-simulation packages are commonly used in engineering. They allow us to investigate behaviour of products and surroundings, focusing on particular areas of physics. Examples are simulations of rigid-body dynamics and kinematics [4] and finite-element simulations to investigate various physical phenomena [5]. A forerunning paper describes in more detail how differences between the modelling techniques underlying the various simulation approaches impede the realization of integrated artefact simulation approaches [6]. Key challenges are the integration of physics phenomena into multiphysics simulation, and the integration of discrete and continuous simulation into hybrid simulation.

In another forerunning paper, approaches for the simulation of human aspects, with or without artefacts, have been reviewed [7]. Human behaviour is usually investigated using ergonomics-oriented software, which typically focuses on aspects of kinematics, field of view and static loads [8]. Simulations that include other aspects of human behaviour, such as cognition and motion control, can be found in areas other than product design or in specialized areas of product design.

The work described in this paper aims at integration of (i) artefact/engineering aspects and human/ergonomics aspects, (ii) the various different types of artefact behaviour and (iii) the various different types of human behaviour in simulation. In a

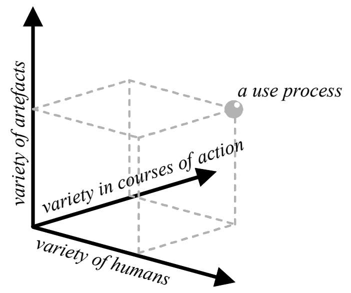


Fig. 2. *The three dimensions of variety in use processes.*

typical use process, many of these aspects and types of behaviour come together. The objective is that knowledge that can be handled by separate existing forms of support is dealt with by one form of support.

1.2. The diversity of use processes

Another knowledge issue that is not resolved by binging together the capabilities of existing simulation approaches is the fact that every use process is different, even if the human user and the product are the same. The variety in use processes has three dimensions (Fig. 2): (i) variety in artefacts, (ii) variety in humans and (iii) variety in the courses of action. The variety in artefacts appears primarily due to the diversity of products and the variety of surroundings in which a particular product can be used. The variety of humans manifests itself in size and dimensions, as has been the subject of intensive study in the area of anthropometrics [9], but there is also a wide variety in the capabilities and habits, of which the effect on the variety of use processes is interlinked with the third dimension.

The third dimension of variety concerns the different possible courses of action. This has been elaborated by researchers who have characterized use processes as problem-solving processes, i.e., humans use products to solve problems. Newell and Simon [10] introduced a theory of human problem solving that

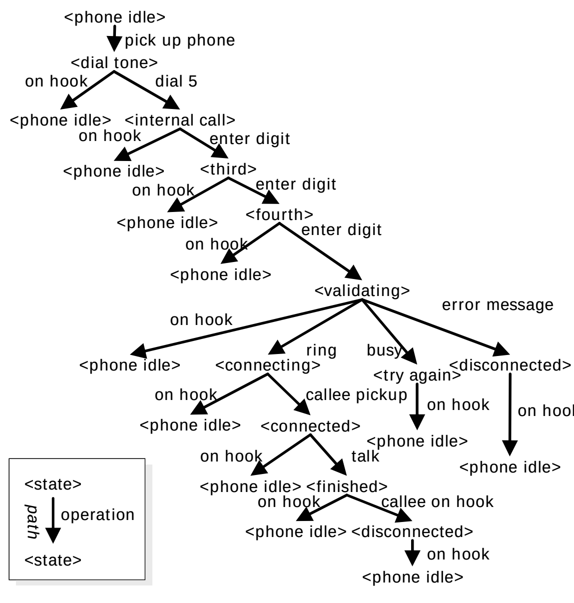


Fig. 1. *Use of a telephone: scenario tree for a particular task (adapted from [13])*

is nowadays widely accepted in cognitive psychology [11]. Its application to the use of products has been explained as follows by Stanton and Baber [12]. The goal of use is to reach a solution to a problem, and the process of solving is called ‘task’. The task is performed in a way analogous to moving through a maze, from the initial state to the goal state. Each junction has various paths representing state-transforming operations. From each junction where the product user arrives one operation is selected, its execution causing change of the present state. Such a maze has also been called ‘scenario tree’ (Fig. 1) because it represents multiple *scenarios* for a given user with a given task in given surroundings, a scenario being one possible way to use the product, i.e., one course through the maze [13].

Selection between available operations is done based on perception and exploration. The available operations appear to the human decision-maker through ‘affordances’ offered by the product[1] [14]. Since various paths are possible from each junction, multiple use processes are possible from a given initial state, some of which may not lead to a solution of the problem. Such unwanted result of a use process is sometimes attributed to ‘human error’. According to Reason [15] it should however be attributed to the design, which has permitted the erroneous operations in the first place. Therefore, an important subgoal in designing for use must be to anticipate all possible forms of use – including ‘errors’, so that the user does not have to adapt his way of using a product to the shortcomings of its design.

The work in this paper aims at making it possible for product designers to generate simulation-based what-if studies and to explore multiple possible scenarios of use by playing with different options for the surroundings, different anthropometric characteristics of humans and different human decision-making patterns.

2. RELATED WORK

Several approaches for combined modelling and simulation of human and artefact behaviour have been proposed and put into practice over the last decades. Approaches to modelling and simulation of human information processing together with physical behaviour are typically tailored to a specific application. For instance, in [16] a simulation approach is presented for human steering control during bicycle-riding for analysis purposes. However, the model of human control behaviour is qualitative rather than quantitative and the approach does not include a provision to replace the bicycle with a different product, for instance from a CAD file. A more versatile approach is offered by the commercial software package Endorphin by NaturalMotion Ltd.[2], for simulating rigid-body mechanical behaviour of human manikins and artefact models that can be imported from CAD. Simulation of human information processing only covers low-level motion control based on genetic algorithms developed by Reil and Husbands [17]. It does not include cognitive decision-making.

There are various approaches that can be considered related because they use finite state machines to represent human and/or artefactual information processing. The most common application field for these approaches is man machine interaction, where finite state machines are used to model and simulate information-processing behaviour of machines (e.g., [18], [19]) and/or human cognition and control (e.g., [20], [21]). However, these approaches do not cover modelling and simulation of continuous physical processes.

Finally, the approach presented by Martins et al. [22] must be mentioned, which is based on finite state-machine modelling and simulation of human decision-making. Continuous behaviour can be included in the form of predefined sequences of animation frames, rather than that it is based on knowledge of the laws of physics.

3. CONCEPTUAL ELABORATION OF THE HUMAN-PRODUCT-SURROUNDINGS SYSTEM AS A HYBRID SYSTEM

Fig. 3 shows the block diagram I propose as a reasoning model about the use process as a process in which humans and artefacts (i.e., products and surroundings) interact. It is based on the assumption that a HPS system can be considered a hybrid system, i.e., a system in which the various behaviours are investigated either as they can be *physically observed* as energy flows, or as they can be *interpreted* as information flows, disregarding the physical phenomena on which they are based[3]. Observed physical behaviour and interpreted physical behaviour are also known as continuous behaviour and discrete behaviour, respectively [23]. In the figure, the two types of flows related to these behaviours are depicted by different types of arrows.

The concept underpinning my method and system framework is the concept of a resource-integrated human-product-surroundings scheme. A resource-integrated model is a combination of (i) a nucleus-based object model (NBOM) of the HPS system that is used for the simulation of continuous behaviour and (ii) a use state machine (USM) which is a behavioural model that is used for the simulation of discrete behaviour.

3.1. Concept of continuous-system modelling

The NBOM that is used for continuous simulation is a composition of nucleus-based models of humans, products and surroundings. Nucleus-based model-

[1] For instance: when seeing a staircase, humans perceive the affordance of climbing
[2] www.naturalmotion.com

[3] For instance, if voltages produced by a device are assumed to represent either ‘0’ or ‘1’, the output ‘100111’ is the abstraction or *interpretation* of a series of output voltages. The physical phenomenon of electric charge is typically disregarded after interpretation.

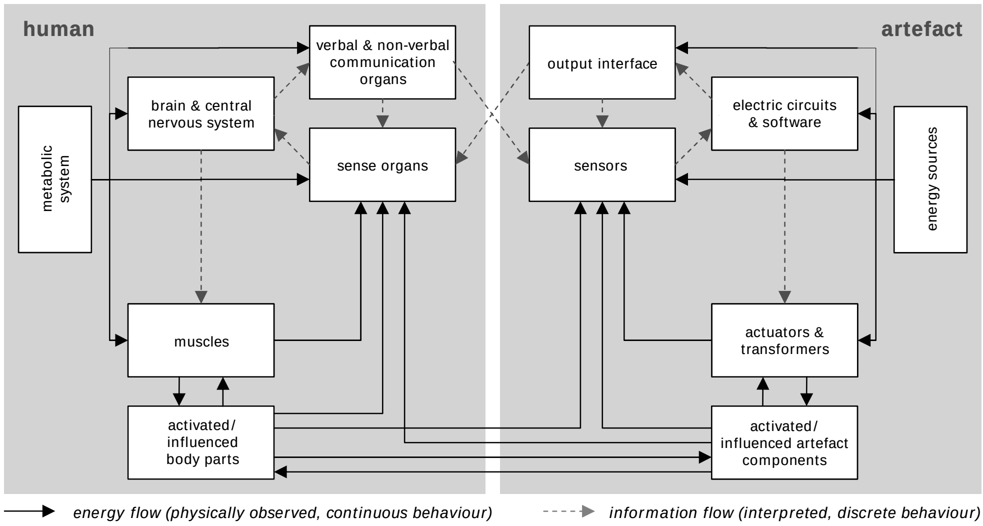


Fig. 3. *My reasoning model: a simplified representation of a human-artefact system and the interaction between its components. It can be considered an adaptation of the human-machine interaction diagram in [2]*

ling ([24], [25]) allows geometric and physically-based representation of humans and artefacts in the same manner and thus makes it possible to simulate their observed physical behaviour integrally. This continuous simulation is based on knowledge about physics, which is built into the models during their creation and instantiation. In a nucleus-based model of an HPS system, the built-in knowledge provides algorithms to calculate the observed physical behaviour from input values, which the NBOM receives during simulation. Input values for two types of parameters are distinguished: (i) *initial parameters*, which define the situation in which the HPS system is at the start of the simulation and (ii) *actuator parameters*, which are imposed upon the NBOM during the course of the simulated use process by the USM (see next paragraph). In a typical use process the actuator parameters refer to displacements, angles and forces exerted by human actuators or *muscles*, which are controlled by the human brain, and by *artefactual actuators*, which are controlled by digital circuits and software in products and surroundings.

3.2. Concept of discrete-system modelling

The USM is a *finite state machine* that represents interpreted physical behaviours ('information processing') performed by the HPS system. A finite state machine is a mathematical construct to describe behaviour of discrete systems. The behaviour is discretized into in states, each of which describes the system for an interval of time [26]. A change between states, called a transition, occurs if the state machine receives specified input [27]. With transitions or states output may also be associated [28]. Representation forms commonly used for state-machine modelling are, among others, state transition diagrams [29], Petri nets [30] and statecharts [31].

In use processes we distinguish two forms of interpreted behaviour that can be simulated as USMs using discrete-simulation algorithms. The first one is decision-making performed by the human brain. Based on what the sense organs perceive around the human, the brain decides to activate muscles as well

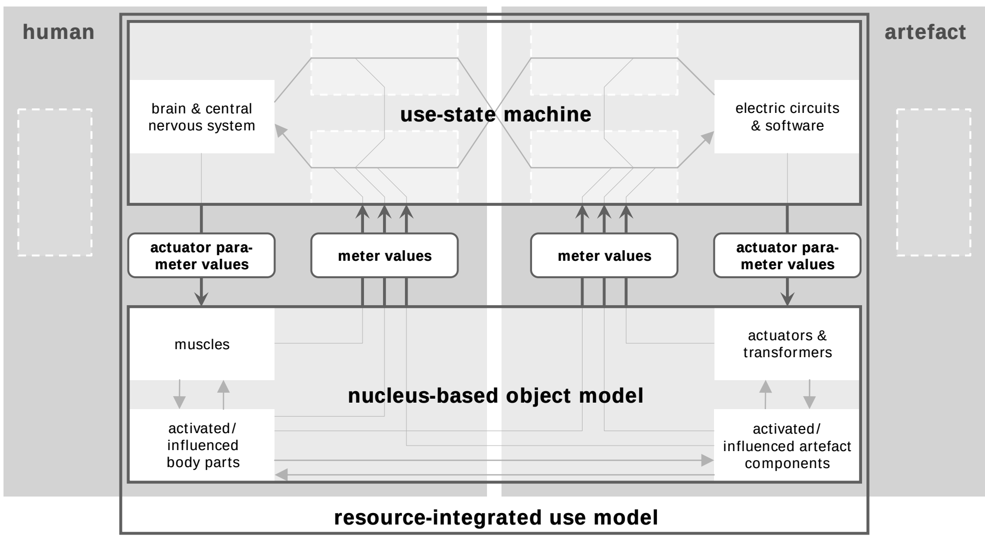


Fig. 4. *Exchange of simulation variables between the USM and the nucleus-based virtual model of the HPS system system shown in Fig. 3.*

as verbal and non-verbal communication organs. Actually, modelling brain functions has been the original purpose of developing the state machine concept in the 1940s [32]. The second form of interpreted physical behaviour that manifests itself in use processes is the behaviour performed by software and digital circuits in products or in the surroundings (Fig. 3). Applying logic to input that is received through input interface elements, these components exert control over actuators and output interface components. Those parameters in the virtual continuous system of which the values are 'read' by the USM are called *meters* and their values are called *meter values*, in accordance with the terminology used in commercial continuous-simulation software packages. This is based on the analogy of attaching a meter (e.g. a tachometer, manometer, etc.) to a real system.

3.3. Concept of hybrid simulation

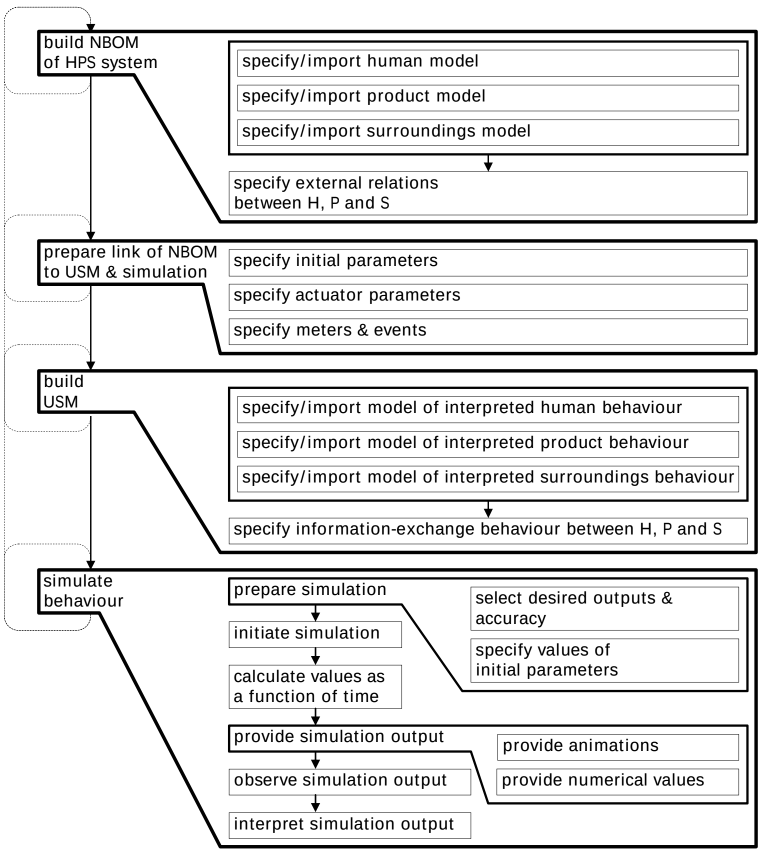
Hybrid simulation of the HPS system combines simulation of observed and interpreted physical behaviour. Continuous simulation of observed physical behaviour and discrete simulation of interpreted physical behaviour are running in parallel, while exchanging values of variables. The USM reacts on specified changes in meter values received from the continuous simulation, which are called *events*. In the discrete simulation these events cause state transitions in the USM as is specified in the behavioural USM model. For given states or state transitions, the USM transfers actuator parameter values to the continuous simulation to control muscles and actuators.

Fig. 4 shows the exchange of simulation variables between a USM and a nucleus-based virtual model of the HPS system according to the above description. Together, the NBOM and the USM are called *resource-integrated use model*. In Fig. 4, the arrows correspond to variables, of which the values are calculated and exchanged by the separate simulation engines predicting the behaviours of the USM and the NBOM, respectively. Comparing the simulation model with the reasoning model in Fig. 3, the following further simplifications were made:

– The behaviour of energy sources (human metabolic system, batteries, etc.) is not simulated. The assumption is that there is sufficient energy to make the other components operate. When needed, a power failure on the artefact side can be modelled as a sensor input (meter value) that causes the USM to shut down.
– Sensory input to electronics/software and to the brain is bypassed in the simulation. Generally, sensors and sense organs can be considered filters determining which signals (meters) from the outside world are registered by electronics/software and the human brain. According to this simplification it is assumed that no filtering is applied, so that electronics/software and the human brain have all the necessary information available to make decisions. It means that the process of translating physical variables to interpreted variables, i.e., to information, is not simulated.
– Processes performed by output interfaces of electronics/software and by human communication organs are not simulated. If artefacts send information to humans through output interfaces (for instance, displays) it is assumed that the information communicated through the interface arrives at the human brain without change. Likewise, if humans send information to artefacts by using communication organs (for instance speech that is to be recognized by speech recognition software) it is assumed that the communicated information arrives at the electronics/software without change. According to this simplification, the process of translating interpreted variables to physical ones (and then back to interpreted variables, in accordance with the previous simplification) is not simulated.

4. METHOD OF MODELLING AND SIMULATING USE PROCESSES

The basic activities in use-process prediction are (i) NBOM modelling, (ii) preparing the NBOM for linking



–

Fig. 5. *Basic activities in use-process modelling and simulation*

with the USM and the simulation, (iii) USM modelling and (iv) hybrid simulation. Fig. 5 shows the flow of activities. First a NBOM is built, consisting of three parts representing the human, the product and the surroundings. The partial models are either built from scratch or selected from previously saved models. There is no prescribed order for the creation or the selection of the three models. If the three models have been defined, the relationships among them can be modelled to create a system-level NBOM, which includes knowledge about the system's observed physical behaviour. The second main step is to prepare this model so that it can be linked to the USM and it can be simulated. When in the next step the USM is built, it must be known to which actuator parameters, meters and events it can refer. And after that, if the simulation is prepared, it must be known for which initial parameters values have to be set. The third step is to build the USM by specifying the interpreted behaviours of the human, the product and the surroundings. If either the NBOM or the USM needs inputs that cannot yet be derived from the outputs of the other model, iterations between the first three main steps are needed. The fourth and final main step is to simulate the integral behaviour of the HPS system. Parameters that determine the simulation accuracy are entered, and the desired presentation form of the simulation output is selected. After the values for the initial parameters have been defined, the simulation is initiated and the values of the variables are calculated as a function of time. The predicted behaviour is shown as an animated HPS system and/or numerical values/graphs of selected values, depending on the choices made in the first simulation step. The simulation output is observed and finally interpreted. After simulation, the NBOM and/or the USM can be revised before another simulation run takes place. Typical reasons for revising the NBOM are (i) to improve the product concept based on the previous simulation outcomes, (ii) to select other virtual humans with different characteristics to check the use process with various users or (iii) to select or model other surroundings to check the use process under different circumstances. Typical reasons to revise the USM are (i) to define a different state machine that represents a different pattern of human decision making to be tested with the product concept, (ii) to revise the existing state machine to include different or new human decisions in order to refine the previous pattern based on the previous simulation run or (iii) to revise the state machine that represents the software or electronics in the product, based on the previous simulation result.

5. EMBEDDING THE WORKFLOW IN THE DESIGN PROCESS

The NBOM and the USM have been devised to offer enhancements to conventional workflows and means of modelling and simulation, with the objective to improve the anticipation of the use process.

The NBOM enriches the conventional workflow in three ways. First, it offers the possibility to model the product with the human and the surroundings as one system. Consequently, the observed physical behaviour of this system can also be simulated as that of one system. Secondly, simulation of nucleus-based models has the advantage over most of the conventional simulation methods that it can handle multiple different physical phenomena simultaneously. Thirdly, apart from these integration advantages, nucleus-based modelling enhances the modelling workflow in conceptual design by allowing the designer to work with imprecise and incomplete models.

Nucleus-based models can be further edited by the designer during the later stages of design up until detail design and manufacturing preparation; for the creation of manufacturing drawings, CNC manufacturing instructions or rapid-prototyping instructions, the nucleus-based product model can be converted to conventional representations, such as STL.

The USM enriches the conventional workflow in two ways. Firstly, it can be seen as an extension of the functional specification of the product, which is typically drawn up at the beginning of the product development process and refined during the subsequent stages of design. While the functional specification is only a description of the intended operation of the product, the USM can be deployed (i) to specify the intended (or expected) behaviour[1] of the user and the surroundings, and perform simulations to check if these behaviours actually make the product operate as intended and (ii) to specify unintended behaviours of the user and the environment, and perform simulations to study their effects. Both applications of the USM assist the designer in improving the product concept.

The second enhancement that the USM brings to the conventional workflow is that it allows modelling and simulation of that part of product operation that is caused by its interpreted physical behaviours. Modelling and simulation of such behaviour is long-time common practice in the development of information systems (see section 2). Resource-integrated modelling gives the opportunity to simulate this behaviour simultaneously with the observed physical behaviour of HPS systems.

6. THE NUCLEUS-BASED MODELLING CONCEPT

Nucleus-based modelling is a next generation of constraints-based parametric modelling that covers both geometric and behavioural aspects throughout the phases of the design process, starting with conceptual design. The nucleus is introduced as a generic modelling entity, which includes two regions

[1] Actually the USM can only be used to model interpreted physical behaviour. The purely physical behaviour – whether it is as intended or not – directly follows from the continuous simulation of the nucleus-based NBOM.

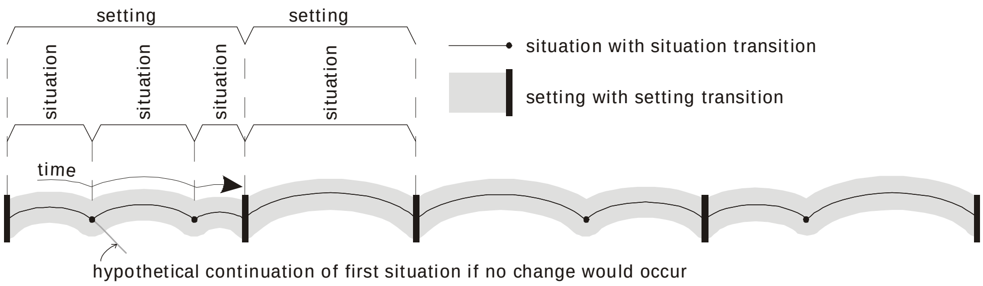


Fig. 6. *Transitions in a use process*

of one or two objects that are interconnected by a system of relations in a particular situation. The idea is that relations, and not objects or entities, are the elementary structures to which systems can be reduced. Relations are existential, manifestation and behavioural associations, dependencies and interactions between humans, products and surroundings. Each subsystem of the HPS system – e.g., the product – is conceived as a purposeful composition of specific instances of nuclei. The nucleus can be instantiated at multiple levels such as particle, component, subassembly and assembly. A predefined set of relations expresses qualitative and quantitative associations, dependencies and interactions between objects in a parameterized form on these levels. As a modelling entity, the nucleus offers advantages in multi-aspect way of conceptual modelling by integrating geometric, structural, physical and behavioural modelling. This paper focuses on the USM; for a detailed elaboration of nucleus modelling the reader is referred to [25].

7. ELABORATION OF THE USE STATE MACHINE CONCEPT

Just like every other process the use process is a sequence of changes in a system. Simulation is performed to predict the changes based on a model of the system, in our case the HPS system. The continuous simulation of the NBOM can predict continuous changes, which can be calculated based on the laws of physics. Such continuous changes are confined to a *situation*. A situation can be defined as a sub-process completely controlled by a particular subset of the physics laws. Additionally, muscles or actuators may be active during the situation according to a behaviour pattern[1] prescribed from the beginning of that situation. In a use process, situations can end in two ways: (i) through a natural transition to a different set of physics laws and (ii) through a change in the behaviour pattern of muscles or actuators, which are controlled by the human brain or by electronics/software. An example of the first case is the series of transitions a falling object undergoes when it bounces, going through three situations of gravity-controlled motion, then deformation and damping and finally, again, gravity-controlled motion. Despite the observed discontinuities in the process, this type of transitions between situations is determined by the laws of nature. Therefore, a succession of situations going through such transitions can be predicted through continuous simulation in one monolithic run. The second type of transitions, however, needs input from the USM.

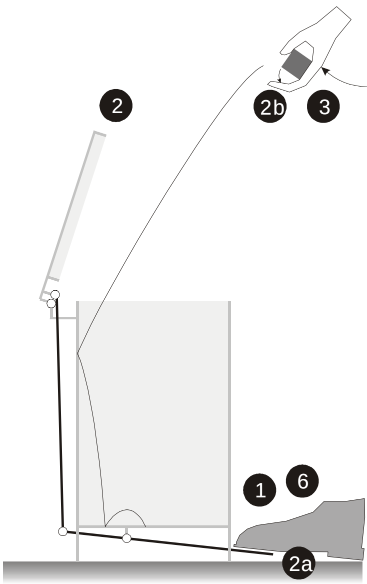


Fig. 7. *Pedal bin*

Fig. 6 presents a graphical interpretation of a use process in which transitions are depicted as 'bumps' in the timeline. A sub-process that takes place from one intervention by the USM until the next intervention by the USM is called a *setting*. A setting consists of one or more situations. During the situations within a setting, but also in the transitions between these situations, the operation of the system can be simulated based on the laws of physics. As a simple example consider a pedal bin used by a human as shown in Fig. 7. At the start of the simulation, the human foot presses down the pedal to open the lid. To simulate this, a motion pattern has been defined in the USM. From the start of this setting (and situation) <1>, the kinetic/kinematical behaviour of the lid-lifting mechanism is calculated by the continuous simulation engine. A next intervention is performed by the USM once the lid has been lifted sufficiently, i.e., the *event* that the value of the *meter* 'angle of the lid' (as is assumed to be perceived by the human's eyes) has reached a particular value <2>. This event triggers human intervention: the USM will stop the displacement of the pedal <2a>. At the same time, the event triggers the USM to make the fingers of the hand, which are holding the garbage object, open up <2b>. During this setting, the situation

[1] For instance, a force exertion as a function of time

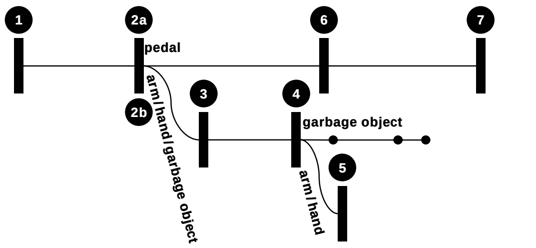


Fig. 8. *Transitions between settings and situations in the intended use process of the pedal bin in Fig. 7*

'garbage object sliding down the hand' takes place, which can be predicted through NBOM simulation. The setting ends when the fingers are fully stretched and therefore stopped by the USM <3>, while at the same time the USM prescribes a throwing motion to move the arm. This setting holds a new situation, in which the sliding of the garbage object in the previous situation is continued, while a throwing velocity is added. At a given moment the USM intervenes by reversing the throwing motion <4> until the arm is back in its starting position <5>. This setting starts with a situation in which the garbage object goes into free fall, followed by successive situations in which it bounces inside the bin, and finally comes to a standstill. This succession of situations can be predicted through NBOM simulation. Once the object enters the bin (an event to be derived from meter values relating the position of the object), the USM prescribes the foot to release the pedal <6> so that the lid is closed. Once the lid is shut the simulation is ended by intervention of the USM <7>.

When we try to depict the example use process in the same fashion as Fig. 6, it appears that situations and settings can be local to a part of the HPS system, while other sub-processes are developing in other parts of the HPS system. For instance, the bouncing of the object inside the bin may continue while the lid is being closed. Apparently, multiple situations and/or settings can be active in parallel; therefore a branched arrangement of settings and situations is needed rather than the linear arrangement of Fig. 6. A branched arrangement that represents the use process described above is shown in Fig. 8. In this figure, the 'bumping' effect of Fig. 6 is omitted for clarity.

Behavioural simulations are applied to predict the course of yet unknown processes. In the use process described above, it is assumed that the litter will fall inside the bin. However, for certain starting positions and motion patterns of the human hand, the result of the continuous simulation is possibly that the litter falls *outside* the pedal bin. If this is the case, the intervention defined for the setting transition at <6> is undefined. The obvious workaround is to define an alternative intervention for the case that the object misses the bin. Fig. 9 shows an extension of Fig. 8, which introduces exclusive branching to include an alternative course of human-decision making belonging to an alternative *scenario*. If at <6> the garbage object crosses the top plane of the

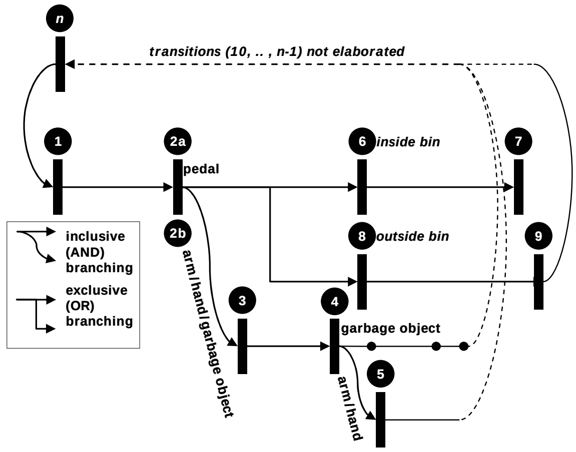


Fig. 9. *Transitions between settings for use of the pedal bin in Fig. 7 including an alternative course of the process in case the garbage object misses the bin*

bin inside the perimeter of the bin *(event)* the use process finishes as described in Fig. 8. However, if it crosses the same plane outside the bin perimeter *(alternative event)*, the alternative path is taken, which eventually loops back to <1> and includes manoeuvring the human hand to the location of the litter beside the bin, picking it up and bringing it back to the throwing location (dashed line, transitions not elaborated in detail). The diagram that connects the transitions has become a network, which is no longer read from left to right only. Arrows have been included to clarify the directions of the various timelines. The similarity between Fig. 9 and common representations for finite state machines, such as Petri nets, is evident[1]. Also, there is a resemblance to the scenario tree in Fig. 1, which is actually also a finite state machine representation. As a generalization, the diagrams in Fig. 1, Fig. 8 and Fig. 9 can be said to be *scenario structures*, of which the tree-shaped variety is a special subclass[2]. If it does not include situation transitions (which are supposed to be handled by simulating the behaviour of the NBOM) a scenario structure can be considered an informal version of a USM. For a USM to be used for computer-based simulation of interpreted physical behaviour and exchange of numerical values with a continuous simulation algorithm, it needs to be formalized. This is elaborated next, based on common finite state-machine conventions.

7.1. Formalization of the use state machine

Finite state machines are commonly used in electronics and software engineering to describe and simulate the intended operation of systems. The USM incorporates the same functionality but it is also used to represent and simulate intended and unintended operation of humans.

A finite state machine describes a system that is always in at least one of a finite set of states. It per-

[1] Note that (just like Fig. 1) Fig. 8 and Fig. 9 are informal representations, which deliberately have not been adapted to a common representation form for state machines. For now, the choice for one of the existing representation forms is considered arbitrary.

[2] In Fig. 1, a feedback loop could also have been created for the 'try again' state.

forms transitions between states as a reaction on inputs that it receives. Since the USM not only receives inputs in the form of events (which are defined based on meter values) but also produces output in order to control actuators, it is a special type of finite state machine, namely a *finite state transducer*[1].

The USM as a finite state transducer can be formally defined as a quadruple *(S, Σ, Γ, T)* where

S is a finite nonempty set called the set of states,

Σ is a finite set called the set of inputs or the input alphabet,

Γ is a finite set called the set of outputs or the output alphabet,

$T \subset S \times \Sigma \times S$ is a nonempty set of transitions ($T : S \times \Sigma \rightarrow S$).

A state corresponds to a setting that takes place in the HPS system. As was explained in the introduction to this subchapter, a setting is a sub-process during which there is no change in the behaviour pattern of muscles or actuators controlled by the human brain or electronics/software[2].

The input alphabet and the output alphabet provide symbols or characters that represent signals, which are exchanged with the NBOM simulation. Elements of the input alphabet and the output alphabet can be concatenated to input strings and output strings, respectively. The input alphabet Σ relates to specified changes that can be detected in the simulated NBOM as meter values, while the output alphabet defines behaviour patterns for muscles and actuators. Based on the input signals a transition from one given state to another given state takes place to enable a new setting, while at the same time a new output signal is transmitted to the NBOM simulation.

Transitions are triggered by events that are defined using the input alphabet. An event has no duration: it corresponds to a value of a given meter variable in the HPS system simulation crossing a given threshold value. Depending on whether the threshold value is crossed while increasing or decreasing, *rising-edge events* and *falling-edge events* are distinguished, respectively. A rising-edge event is written as $E_R = \uparrow v(x)$, where v is the variable of which exceeding value x causes the event. A falling-edge event is written as $E_F = \downarrow v(x)$, which corresponds to the event of v falling below x. As an integral part of an event, logical conditions may be added that must be true while the event happens for the transition to take place. Such logical conditions are expressed as inequalities containing (combinations of) meter variables. Generating events from the continuous NBOM simulation output is a pre-processing step, which is performed outside the USM.

[1] en.wikipedia.org/wiki/Finite_state_transducer
[2] Note that unlike the term might suggest, a 'state' does not correspond to a 'static' stage in the use process. The HPS system can change during a state (i.e., during a setting), but the changes are completely defined by physics laws and by predefined behaviour patterns.

Formally, a transition $\tau \in T$ is defined as
$\tau_i\,(s_i, \sigma_i) = s_j$ with

$s_i \in S$ the input state,

$\sigma_i \in \Sigma$ the input event associated with the transition, and

$s_j \in S$ the output state.

The output alphabet Γ contains the symbols that are needed to represent the signals the USM transfers to the NBOM during simulation. A signal specified using the output language defines a control instruction for an actuator A by defining its actuator parameter as an input variable y_A in the time domain, $y_A = y_A(t)$. Actuator parameters are typically forces, torques, displacements, velocities, accelerations, etc. A control instruction (for short 'control') is used to prescribe a behaviour pattern for the actuator. It is valid *from* entering the state to which it is associated *until* another control instruction for the same actuator is transferred from the USM to the NBOM at the entry of another state. A control instruction is defined in the USM using a common continuous function description, e.g., the output string "$y_A = c_1 + c_2 \cdot \cos(t)$". A control instruction may be an empty string.

7.2. Concept of discrete-event based simulation of the use state machine

In resource-integrated simulation we distinguish simulation time and calculation time. Simulation time corresponds to happenings in the virtual HPS system that is being simulated, while calculation time corresponds to happenings in the system that performs the simulation.

When, in calculation time, the USM receives an event it evaluates if the event is associated with any outgoing transitions of states that are active at that point in time. For all combinations of outgoing transitions and events for which this is true, the concerned transitions are taken and the USM assumes the states for which these are the ingoing transitions. Consequently, the actuator control instructions associated to these states are transferred to the NBOM simulation. If an incoming event is not associated to any outgoing transition of current states, the USM does not react.

From the moment the USM receives an incoming event until it outputs the actuator control instructions that follow from it, simulation time does not advance. From the moment a new set of states is entered until a next event comes in, simulation time advances as long as the physical processes simulated outside the USM need until they trigger the occurrence of the next event. Thus, the simulation time between events is completely determined by the continuous simulation.

The first event in use-process simulation is the start event which is generated by the system user. It triggers the system to start the simulation of the USM.

Simulation stops if a final state is reached that is defined by the user in the USM modelling activity.

8. SYSTEM ARCHITECTURE AND SOFTWARE COMPONENTS

A system that supports creation and simulation of resource-integrated use models should offer modelling functionality, simulation functionality and user-interface functionality. Fig. 10 shows how the functionality of the proposed system is further decomposed. Modelling involves NBOM modelling, USM modelling and defining the connections between the NBOM and the USM so that their behaviours can be simulated as the behaviour of one hybrid system. Simulation involves physical behaviour simulation, interpreted-physical behaviour simulation and connecting the two simulations by generating events and controlling actuator parameters. The user interface functionality enables the user to provide input for modelling and to initiate simulation runs, and provides output to the user in the form of modelling feedback and simulation results.

The proposed system architecture is subdivided into three modules, (i) the nucleus module, (ii) the hybrid resource integration module and (iii) the use state machine module. Each of these modules deals with different aspects of the modelling, simulation and user-interface functionality of the system as shown in the matrix in Fig. 11.

9. TENTATIVE PILOT IMPLEMENTATION OF THE USE STATE MACHINE AND HYBRID RESOURCE INTEGRATION MODULES

Based on existing commercial software packages a tentative system was implemented to test the concepts of modelling and simulating use state machines and linking them to continuous simulation. The implementation of nucleus modelling and simulation was not yet included because it requires additional programming effort, which has been planned as a next step. As a provisional substitute for nucleus-based object modelling and continuous simulation MSC visualNastran4D (vN4D) was used, a commercial software package for modelling and simulation of kinetic mechanical behaviour in systems consisting of rigid-body components.

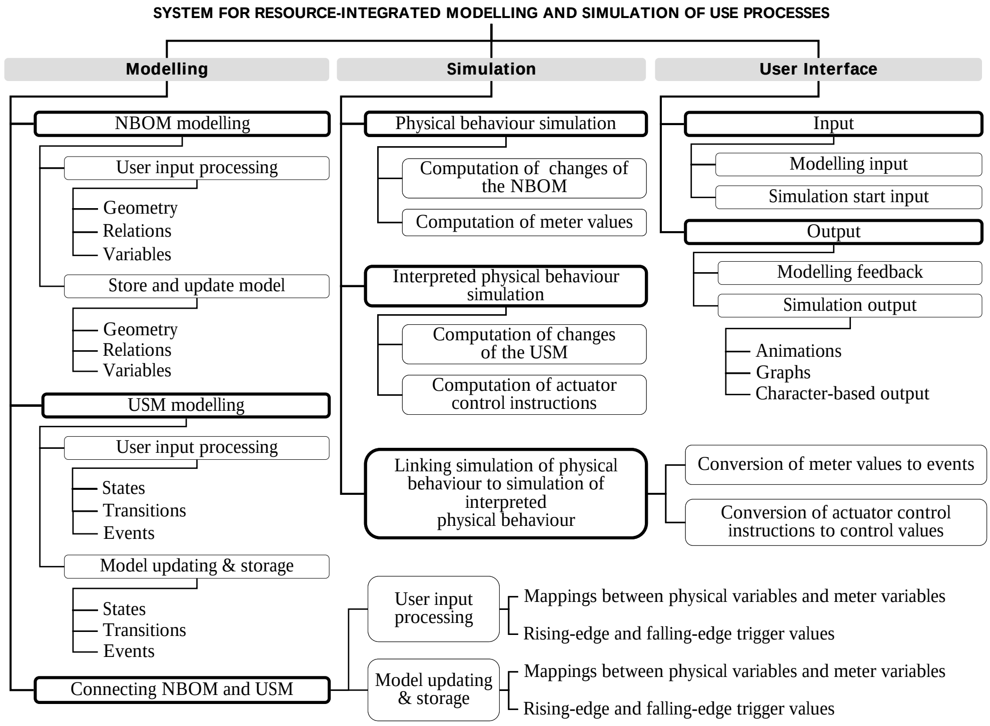


Fig. 10 *Breakdown of the system functionality*

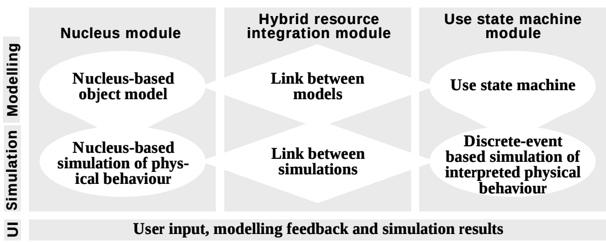


Fig. 11 *Overview of the concept of the system architecture: main software components and functions. (note: 'UI' = user interface)*

Two components of the commercial MATLAB software package could be used to model and simulate the USM and the hybrid resource integration: (i) Simulink Stateflow and (ii) Simulink elements for building conventional block diagrams, respectively.

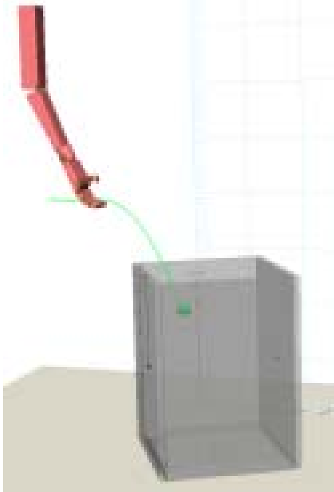


Fig. 12 *Human-product-surroundings system that was used to test the tentative pilot system, modelled and simulated with MSC VisualNastran4D.*

Error! Reference source not found. shows the very simple human-product-surroundings[1] system that was simulated to test the tentative pilot system: a human arm attempting to throw a piece of litter into a litterbin. The model shown in the picture is further discussed in 9.2.

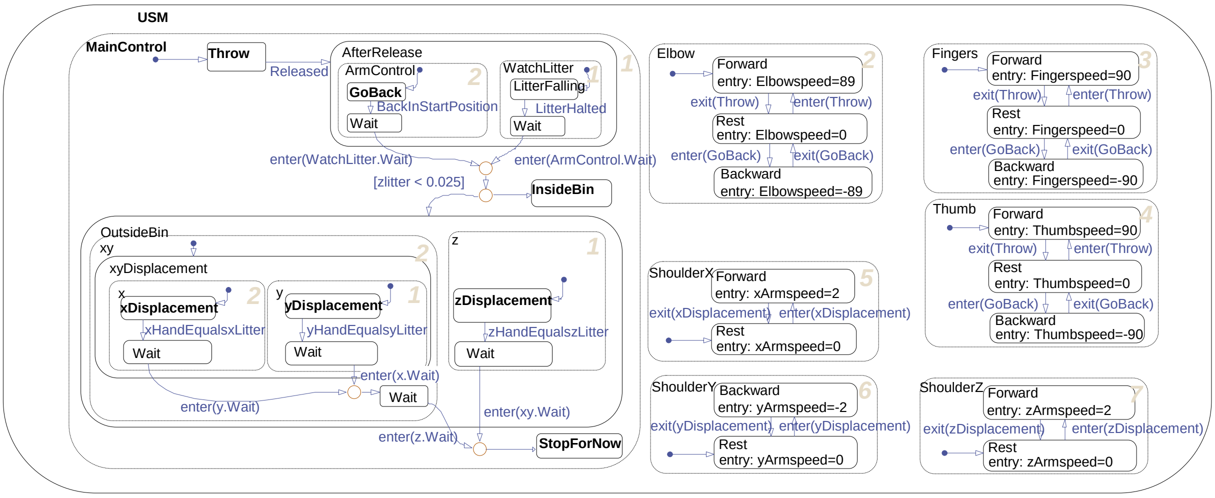


Fig. 13 *Use state machine modelled with Simulink Stateflow*

9.1. The use state machine

Fig. 13 shows the USM that was created with Simulink Stateflow to describe how human decision-making controls the motions of the human arm. Stateflow uses a variant of the statechart notation established by Harel ([31], [33]). Fig. 14 explains some key notations used in Stateflow diagrams. As an extension to the formal description in 7.1, transitions can not only be triggered by external events received from the continuous simulation but also by events within the USM, such as entering or exiting a state in a parallel sub-process. Internal events are not strictly needed for building USMs, yet they offer a convenient means to create modularity among sub-processes, which is an opportunity specifically offered by Stateflow diagrams and statecharts. Another aberration from the formal state-machine description in 7.1 is that logical conditions evaluating values of meters can be formulated independently and connected to transitions within the USM, rather than that they have to be part of the definition of events prepared outside the USM. This opportunity offered by the Stateflow notation is used to reduce the number of separate events that would otherwise have to be defined as a part of the hybrid resource integration outside the USM[2]. The disadvantage is that the USM itself needs to have access to the values of the meters referred to in conditions.

At its highest level the USM in Fig. 13 contains seven sub-processes that can run in parallel. These are represented by dashed containers inside the outermost bounding box. The sub-process named ‘MainControl’ describes the main decision-making process that starts with throwing (corresponding state: ‘Throw’). If the litter has been released from the hand and has reached a certain distance from the fingertips, the arm goes back (corresponding state: ‘GoBack’) until it has reached its original position. If the litter lands inside the bin, the end state ‘InsideBin’ is reached. If the litter has landed outside the bin and has almost stopped moving, the hand moves to the position of the litter as a preparation for picking it up and trying again with a different throwing speed (corresponding states for moving along three axes: ‘xDisplacement’, ‘yDisplacement’, ‘zDisplacement’). Modelling and simulating the process of picking up turned out to be difficult as the

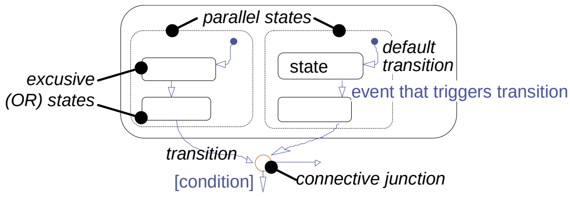
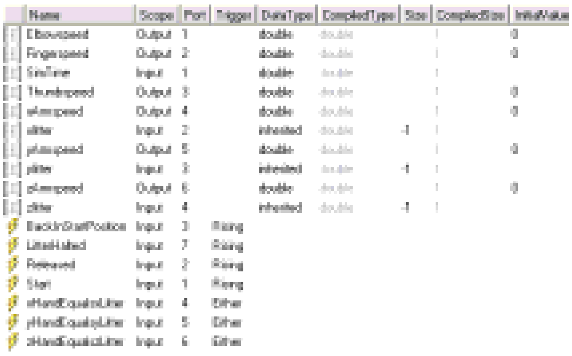


Fig. 14 *Legend of notations used in Fig. 13.*

[1] *human* = arm + hand; *product* = litterbin; *surroundings* = floor + litter

[2] For example, a definition of an event according to the formalization in 7.1 would be “$E_R = \uparrow v(0)$ AND $x<5$”. Instead, the event is now defined only as “$E_R = \uparrow v(0)$”, while a transition in the USM that is associated with E_R has the additional condition “$x<5$”. The E_R, which

continuous simulation algorithms in vN4D do not support investigation of flexible deformations in the hand. Therefore, the use process ends here with the ‘StopForNow’ state. A retry loop for throwing at a different speed can be included once nucleus-based simulation of continuous behaviour has been implemented and gripping behaviour of the hand can be modelled and simulated more adequately. Beside the ‘MainControl’ process, there are six modular sub-processes that can run parallel to it. They describe activation of rotations of the lower arm, the fingers and the thumb as well as translations of the shoulder in more detail. They are linked to the ‘MainControl’



Name	Scope	Port	Trigger	DataType	CompiledType	Size	CompiledSize	InitialValue
Elbowspeed	Output	1		double	double		1	0
Fingerspeed	Output	2		double	double		1	0
SimTime	Input	1		double	double		1	
Thumbspeed	Output	3		double	double		1	0
xArmspeed	Output	4		double	double		1	0
xlitter	Input	2		inherited	double	-1	1	
yArmspeed	Output	5		double	double		1	0
ylitter	Input	3		inherited	double	-1	1	
zArmspeed	Output	6		double	double		1	0
zlitter	Input	4		inherited	double	-1	1	
BackInStartPosition	Input	3	Rising					
LitterLifted	Input	7	Rising					
Released	Input	2	Rising					
Start	Input	1	Rising					
xHandEqualsLitter	Input	4	Either					
yHandEqualsLitter	Input	5	Either					
zHandEqualsLitter	Input	6	Either					

Fig. 15 *Definition of events (thunder symbols) and other variables for the* USM.

sub-process through internal events.

Fig. 15 shows the definition table for events and variables that is used to connect the USM to the hybrid simulation.

9.2. Continuous simulation model

The object model for simulation of the continuous mechanical behaviour of the HPS system shown in **Error! Reference source not found.** was created using the basic solid-modelling functionality built into the vN4D software. Apart from rigid bodies representing the geometry of the HPS system, other elements that have been defined in vN4D are joints connecting human limbs, actuators representing muscles and controls and meters for linking to the discrete simulation. Controls are inputs to be defined for each actuator, and meters are outputs that can be defined as distances between specific points on rigid bodies, angles between bodies, velocities, etc. A so-called *vN4D plant* is defined to link a Simulink model to a vN4D model through a user-specified selection of its controls and meters. Fig. 16 shows the selection of controls and meters that was used in the vN4D plant of the considered HPS system

In simulating the use of the litterbin, it was not the intention to give an accurate representation of human geometry or to achieve realistic simulation of human muscle behaviour and human kinetics. It is assumed that knowledge from other research can be used at a later stage to build more realistic human models. For instance, one of the simplifications that

is defined outside the USM in the link with the object model can now also be used for other transitions with different conditions.

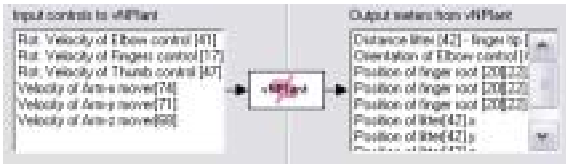


Fig. 16 *Linking controls and meters*

have been made for this proof-of-ideas simulation application has been to model muscles as rotating actuators positioned at joints, instead of linear actuators connected to bones.

9.3. Hybrid resource integration: linking the models

Fig. 17 shows how the link between the models is realized with block-diagram elements (depicted against the shaded background). They connect a block representing the continuous vN4D simulation model (or ‘plant’) to a block representing the USM. A block diagram connects a combination of a particular continuous-simulation model and a particular USM. The linking elements convert output (meter) values of the continuous simulation so that they can be used as input to the USM simulation and they convert output of the USM as control values for the continuous simulation.

Meter values are prepared for use in the USM in two ways. In the first place, falling-edge and rising-edge events are defined based on specified threshold values for (combinations of) continuous output variables. If an event is defined based on a combination of multiple meter values (e.g., the difference between two coordinates, to express a distance), these meters are processed by block-diagram elements that perform the necessary operations on them (e.g., subtraction). After that, the value (of a meter or a combination of meters) is processed by a ‘hit crossing’ block (see Fig. 17) that will send an event with a specified event name to the USM if the value falls below or exceeds a specified value.

In the second place, values of those meters that are used to evaluate logical conditions defined within the USM (or processed combinations thereof) are passed on to the USM directly.

Fig. 17 also contains the start event, which is the only event that is not based on output from the continuous simulation but from the system user.

9.4. Running hybrid simulations

Once the object model and the USM have been modelled and connected, hybrid simulations can be performed by activating the ‘start’ button in the Simulink user interface. During the simulation, vN4D animates the behaviour of the HPS system, while Stateflow animates the USM by highlighting the active states or transitions. By entering different throwing speeds into the USM diagram, different scenarios can be simulated. Some scenarios end with the litter at the bottom of the bin and the arm back in its original position (**Error! Reference source not**

found. shows an intermediate snapshot of such a scenario), while at the end of other scenarios the

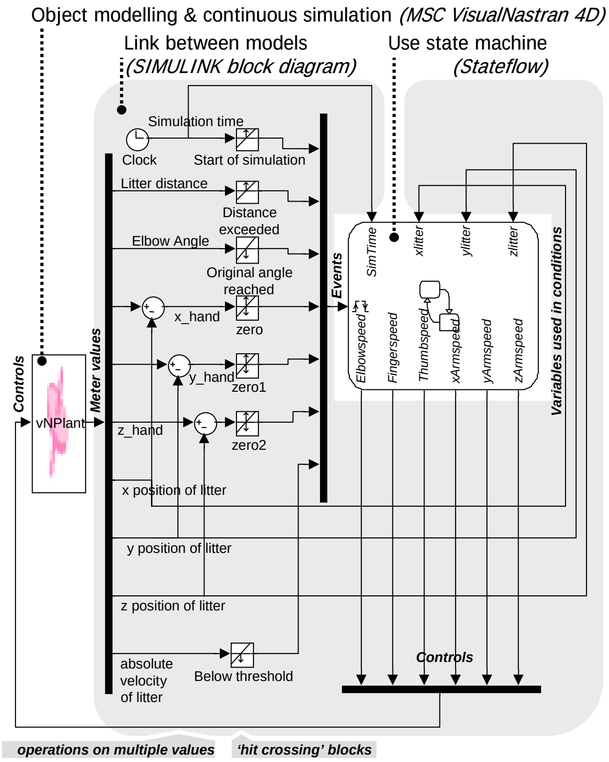


Fig. 17 *Hybrid simulation model*

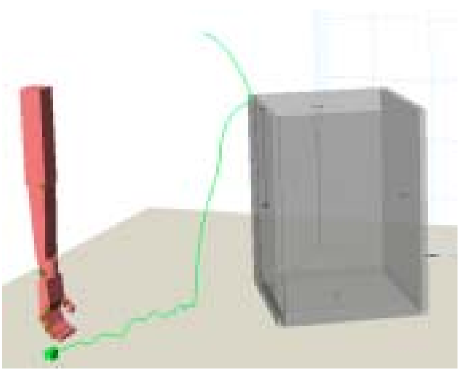


Fig. 18. *End result of simulation if the throwing speed is set too low to get the litter into the bin*

piece of litter is on the floor outside the bin with the arm just above it, ready to pick it up (Fig. 18).

10. DISCUSSION AND CONCLUSIONS

In this paper the concept of use state machines and hybrid resource integration has been elaborated to a level that made it possible to create a hybrid model and test it in a basic application example. In the example a use state machine is linked to a continuous simulation and tested on a use process of a very simple human-product-surroundings system. Of course it would be more interesting to test the concept with more complex products such as coffee makers, which allow for a large diversity of use scenarios with different possible outcomes (either intended or unintended) [34]. It is expected that hybrid simulation of the use of such products will be possible once a full implementation of nucleus-based object modelling and simulation is available, so that a wider range of observed physical behaviours can be included in continuous simulation. For now, the tentative pilot implementation could confirm that it is possible to apply a USM as a networked model of human-decision-making, with exclusive (OR) and inclusive (AND) branching that controls human motion patterns in a continuous simulation, and to link the results of the continuous simulation to the USM so that they influence the decision making. By using exclusive branching in the USM, different scenarios could be simulated with different outcomes.

It can be argued that, even for the simple processes that have been modelled here, the USM shown in Fig. 13 is already quite complex, and that it is questionable whether designers are willing to go through the necessary modelling effort during conceptual product design. However, it has to be noted that the Stateflow representation was chosen mainly based on its availability and on the knowledge that it can easily be linked to other simulations. It may very well be true that other finite state machine representations make it possible to create simpler models for the same scenario structure. For instance, part of the complexity in Fig. 13 is caused by the repetitive use of a "Wait" state that is needed to synchronize outgoing transitions from multiple states. With Petri nets, synchronization can be modelled in a more straightforward way [35].

Also, it must be mentioned that probably not all the possibilities of modelling with Stateflow have been used to their full potential. One idea that seems to be promising is to prepare modular sub-routines of human decision-making and muscle control that can be reused in different simulations, so that the modelling effort can be reduced. Candidate subroutines for such modules would be common motion patterns, such as throwing something in a given direction, arranging the human body into a specified posture, or gripping an object. For the preparation of such modules, additional research into human motion patterns may be needed.

11. FUTURE WORK

Among the further development activities towards resource-integrated modelling and simulation of use processes the development of system components to enable nucleus-based object modelling and simulation has the fist priority. Since the integration between vN4D and Simulink/Stateflow as described in this paper turned out to be successful, the idea is to write a front-end application to adapt nucleus-based models so that they can be simulated in vN4D and obtain a first proof of ideas. The key challenge lies in the fact that nucleus-based object models are particle-based, while vN4D works with rigid volumetric objects. However, rigid objects in vN4D can be connected with discrete flexible components such as springs and dampers. If the front-end application discretizes rigid volumetric objects into particle clouds in which the particles are connected by springs and dampers, it is expected that the particle system can be represented and simulated in vN4D. The result would be a pilot system to simulate all mechanical behaviours in use processes, including large deformations, together with interpreted physics (i.e., information processing and decision-making). One key application of simulating large deformations is to investigate physical interaction between humans and artefacts, where modelling and simulating human body parts as rigid elements is often an inadequate workaround. A dedicated implementation of nucleus-based modelling that will also support observed physical behaviour outside the mechanical domain (thermodynamics, acoustics, etc.) is planned as a later development effort.

References

[1] Korn, G.A., Wait, J.V., *Digital continuous system simulation,* Englewood Cliffs: Prentice-Hall, Inc., 1978

[2] Eggert, R.J., *Engineering Design,* Upper Saddle River: Pearson Prentice-Hall, 2005

[3] Roozenburg, N.F.M., Eekels, J., *Product design: fundamentals and methods,* Chichester: John Wiley & sons, 1995

[4] Wang, S.-L., *Motion simulation with working model 2D and MSC.visualNastran 4D,* Journal of Computing and Information Science in Engineering, Vol. 1, 2001, pp. 193-196

[5] Zienkiewicz, O.C, Taylor, R.L., *The finite element method,* Vol. 1,2,3, Oxford: Butterworth-Heinemann, 2000

[6] Van der Vegte, W.F., *A survey of artifact-simulation approaches from the perspective of application to use processes of consumer durables,* Proceedings of the TMCE, Ljubljana, 2006, pp. 617-632

[7] Van der Vegte, W.F., Horváth, I., *Including human behavior in product simulations for the investigation of use processes in conceptual design: a survey,* Proceedings of ASME-IDETC, Philadelphia, 2006

[8] Chaffin, D.B., *Digital human modeling for vehicle and workplace design,* Warrendale, PA: Society of automotive engineers, Inc., 2001

[9] Pheasant, S., Haslegrave, C.M., *Bodyspace - Anthropometry, Ergonomics and the Design of Work,* London: Taylor & Francis Ltd., 1996

[10] Newell, A., Simon, H.A., *Simulation of human thought,* In: Dutton, J.M., Starbuck, W.H. (eds.), Computer simulation of human behavior. New York: John Wiley and Sons, Inc., 1971

[11] Eysenck, M.W., Keane, M.T, *Cognitive Psychology (4th Edition),* Hove, UK: Psychology Press., 2000

[12] Stanton, N.A., Baber, C., *A systems analysis of consumer products,* In: Stanton, N.A.(Ed.), Human factors in consumer products, London: Taylor & Francis Ltd, 1998., pp. 75-90

[13] Hsia, P., Samuel, J., Gao, J., Kung, D., Toyoshima, Y., Chen, C., *Formal approach to scenario analysis,* IEEE Software, March 1994, pp. 33-41

[14] Gaver, W.W., *Technology affordances,* Proceedings of CHI, New Orleans,1991

[15] Reason, J., *Human error,* Cambridge: Cambridge University Press, 1990

[16] Šuc, D., Bratko, I., *Qualitative trees applied to bicycle riding,* Linköping Electronic Transactions on Artificial Intelligence, Vol. 4, Section B, 2000, pp. 125-140

[17] Reil, T., Husbands, P., *Evolution of central pattern generators for bipedal walking in a real-time physics environment,* IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, 2002, pp. 159-168

[18] Parnas, D.L., *On the use of transition diagrams in the design of a user interface for an interactive computer system,* Proceedings of the ACM 24th National Conference, 1969, pp. 379-385

[19] Glinz, M., *Systematically combining specifications of internal and external system behavior using statecharts,* Proceedings of the 3rd ICSE Workshop on Scenarios and State Machines: Models, Algorithms and Tools, 2004

[20] Rauterberg, M., Fjeld, M., Schluep, S., *Goal setting mechanism in Petri net models of human decision making,* Proceedings of the IEEE Int. Conference on Systems, Man and Cybernetics, Orlando, 1997, pp. 2696-2701

[21] Liu, A., Salvucci, D., *Modeling and prediction of human driver behavior,* Proceedings of the 9th Int. Conference on Human-Computer Interaction, Los Angeles, 2001, pp. 1479-14839

[22] Martins, J.D.F., Nedel, L.P., Menezes, P.F.B., Accorsi, F., *An automata-based animation engine to control articulated characters,* Proceedings of SBC Symposium on Virtual Reality, São Paulo, 2004, pp. 39-50

[23] Bobrow, D.G., *Qualitative reasoning about physical systems: an introduction,* Artificial Intelligence, Vol. 24, 1984, pp.1-5

[24] Horváth, I., Van der Vegte, W.F., *Nucleus-based product conceptualization – Principles and formalization,* Proceedings of the 14th International Conference on Engineering Design, Stockholm, 2003

[25] Rusák, Z., Horváth, I., Van der Vegte, W.F., *First steps towards an all embracing relations-based modelling,* Proceedings of ASME-CIE, Salt Lake City, 2004

[26] Fishwick, P.A., *Simulation model design and execution: building digital worlds,* Englewood Cliffs: Prentice-Hall, Inc., 1995

[27] Khoussianov, B., Nerode, A., *Automata theory and its applications,* Boston: Birkhäuser, 2001

[28] Lee, D.-T., *Evaluating real-time software specification languages,* Computer Standards & Interfaces, Vol. 24, 2002, pp. 395-409

[29] Gill, A., *Introduction to the theory of finite-state machines,* New York: McGraw-Hill book company, Inc., 1962

[30] Petri, C.A., *Nets, time and space,* Theoretical Computer Science, Vol. 153, 1996, pp. 3-48

[31] Harel, D., *Statecharts: a visual formalism for complex systems,* Science of Computer Programming, Vol. 8, 1987, pp. 231-274

[32] Hopcroft, J.E., Motwani, R., Ullman, J.D., *Introduction to automata theory, languages, and computation,* Reading: Addison Wesley, 2001

[33] The MathWorks, Inc., *Stateflow and stateflow coder for use with Simulink - modeling, simulation, implementation,* Natick: The MathWorks, Inc., 2004

[34] Van der Vegte, W.F., Kitamura, Y., Koji,Y. Mizoguchi, R., *Coping with unintended behavior of users and products: ontological modelling of product functionality and use,* Proceedings of ASME-CIE, Salt Lake City, 2004

[35] David, R., Alla, H., *Discrete, continuous and hybrid Petri nets,* Berlin: Springer Verlag, 2005