

Review Article

Achieving Closed-Loop Control Simulation of Human-Artefact Interaction: A Comparative Review

Wilhelm Frederik van der Vugte and Imre Horváth

Faculty of Industrial Design Engineering, Delft University of Technology, Landbergstraat 15, 2628 CE Delft, The Netherlands

Correspondence should be addressed to Wilhelm Frederik van der Vugte, w.f.vanderveugte@tudelft.nl

Received 31 March 2011; Revised 5 August 2011; Accepted 12 August 2011

Academic Editor: MuDer Jeng

Copyright © 2011 W. F. van der Vugte and I. Horváth. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To include user interactions in simulations of product use, the most common approach is to couple human subjects to simulation models, using hardware interfaces to close the simulation-control loop. Testing with virtual human models could offer a low-cost addition to evaluation with human subjects. This paper explores the possibilities for coupling human and artefact models to achieve fully software-based interaction simulations. We have critically reviewed existing partial solutions to simulate or execute control (both human control and product-embedded control) and compared solutions from literature with a proof-of-concept we have recently developed. Our concept closes all loops, but it does not rely on validated algorithms to predict human decision making and low-level human motor control. For low-level control, validated solutions are available from other approaches. For human decision making, however, validated algorithms exist only to predict the timing but not the reasoning behind it. To identify decision-making schemes beyond what designers can conjecture, testing with human subjects remains indispensable.

1. Introduction

For virtual testing of how a product behaves when it is being used, (mechanical) engineering simulation software such as finite element and multibody simulations have become widespread. What these simulation packages generally do not offer is built-in functionality to include human interaction with products. To include this interaction in simulation-based testing, inputs to the engineering simulation must somehow involve signals that represent human actions with the product, which in turn are influenced by input that the human has received from the interaction process. Figure 1 shows a basic reasoning model of human-product interaction with the two principal approaches to deal with human interactions in simulations. It contains similar control loops on both the human and the product side, involving sensing, signal and information processing, and force exertion (obviously, the control loop on the product side is optional, since it is absent in many products that lack embedded control (e.g., chair, pedal bin, and bicycle)). Common engineering simulations typically deal with the central rectangle marked “physical interaction”, whereby the

incoming arrows are defined as “conditions” or “loads”, which can be time dependent but which typically do not depend on feedback from interaction control loops.

One principal approach to include human interaction in simulations is the interactive or human-in-the-loop (HIL) approach, where human subjects are linked to the computer simulation by means of a hardware interface, for instance, virtual reality (VR) and haptics equipment (e.g., [1]). Our research work follows the other noninteractive approach, so that recruiting human subjects and installing hardware interfaces is not needed. In this paper, we have reviewed approaches that can be used for such *software-in-the-loop* (SIL) simulations of human-product interaction. In the case of full SIL simulation, the interaction loop on the human side is closed by a human model that receives input from the product simulation and acts in the virtual space based on these inputs without receiving commands from a human.

By following this approach, we expect to achieve the following benefits for designers. Firstly, deploying human subjects is expensive and time consuming [2] and the hardware interfaces are typically costly and uncomfortable

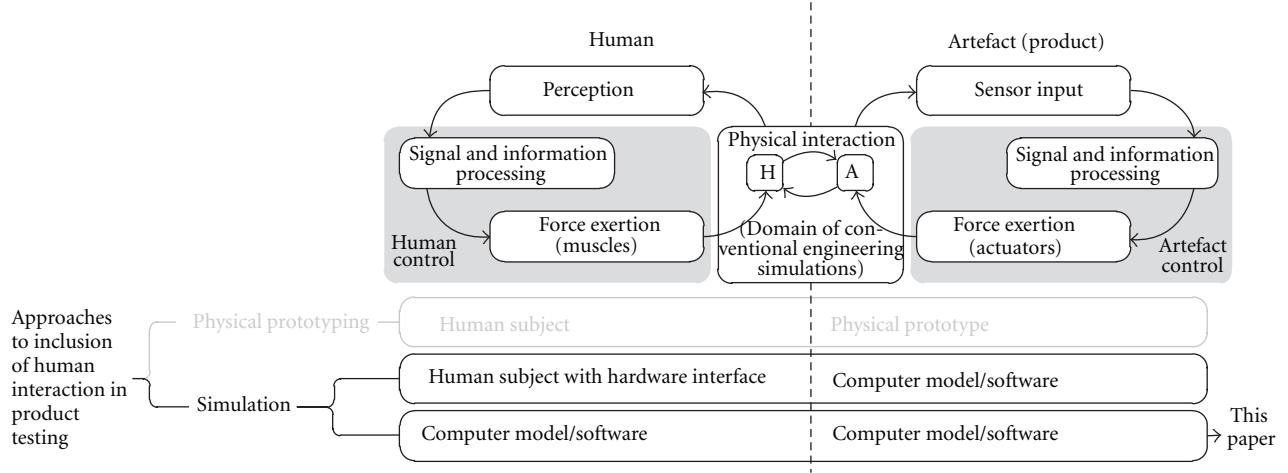


FIGURE 1: Basic reasoning model of human-artefact interaction (top), with different approaches to inclusion of human interaction in product testing—in particular, product simulation.

for the subjects [3]. Although for the final design extensive testing with humans cannot be rendered unnecessary, a cheaper form of testing might make it possible to perform more frequent testing of interaction aspects in earlier stages of design, where evaluation of these aspects is now often neglected.

Secondly, interactive testing with humans in the loop requires that simulations run in real time so that subjects can act naturally. This requirement cannot always be met by typical engineering simulation approaches. Especially with complex product models, computation-intensive approaches such as finite-element analysis are too slow [4]. On the other hand, simulations can sometimes also run *faster* than real time—in particular when simplified models are used, which is common in conceptual design. If these simulations have to run in a real-time environment, the speed advantage cannot be exploited. Faster simulation runs give designers the opportunity to perform more tests, for instance, with variations of a design or with different scenarios. In addition, it can be said that in subsequent tests with variations of a design, there is a risk that human subjects are influenced by their experience with previous variations, whereas virtual human subjects can be “reset” for each simulation run. This makes more objective comparisons possible.

Beside investigating the options to close the loop on the human side, we have also reviewed the possibilities to include embedded control in artefacts, which is an issue of increasing importance considering the current proliferation of electronics, mechatronics, and ubiquitous technologies in interactive products. We found that common engineering simulation software generally has no built-in solution to include embedded control, but that software solutions from control engineering that can be connected to product simulations are widely available and widely used. Because of the analogy between human control and product control as shown in Figure 1, these solutions are also potentially interesting for closing control loops in human interaction, where they are currently only scarcely applied.

This paper aims to (i) present a survey of the state of the art based on the literature in the fields of engineering, ergonomics, human motor science, psychology, and computer science, (ii) indicate directions towards implementation of SIL human and artefact control in simulations of the use of products, and (iii) identify the limitations of SIL simulation. Based on our initial findings [5, 6], a proof-of-concept implementation was actually already presented in [7–9]. However, the control-related survey in this paper has not been previously published in this form.

The remainder of the paper is structured as follows. In Section 2, we have clarified the key terminology related to control in the context of simulations. This includes the important distinction between continuous and discrete or logical control. In Section 3, we discuss the various forms of control in artefacts and humans, and we discuss the reasoning models about human control behaviour from literature on human motor science and cognitive psychology. Section 4 discusses approaches for inclusion of continuous control in artefacts and humans. Section 5 addresses logical control and how it has been applied in simulations of artefact systems and humans, respectively. Section 6 further elaborates on the application of scenarios as the key concept for simulations based on conjectured human decision making, which have been also implemented in our SBIS approach that is outlined in Section 7. In Section 8, the findings are discussed based on a systematic comparison of the most relevant approaches, and Section 9 wraps up with the conclusions and suggestions for further work.

2. Terminology

2.1. Simulation, Emulation, and Conjectured Instructions. This paper is about closing control loops in simulation of human-artefact interaction fully based on software. In principle, there are four types of software algorithms that can be deployed for generating actions or operations performed by

humans or artefacts:

- (i) software based on true simulation models. By “simulation models”, we mean models that have been scientifically validated, that is, based on the laws of nature or on results from empirical research,
- (ii) software executing instructions that unambiguously correspond to the (known) instructions that effectuate the actions or operations. This is also called *emulation* [10],
- (iii) software that executes instructions that have been conjectured by the software user as expected actions,
- (iv) a combination of the above.

Of the three standalone options, true simulation is the ideal solution, because it produces unbiased predictions. Engineering simulations, which form our starting point (central rectangle in Figure 1), belong to this category. Processing of actions and operations in the loops on both sides of the central rectangle, which we want to add in order to deal with the control-related functionality, should preferably also be based on simulation. Here, emulation, which is typically applied to programmed instructions in artefacts, may be considered as a second best option: it can also be regarded as realistic, provided that the physics behaviour of the hardware components is not critical (it can be critical, e.g., in the case of multiple devices communicating over a wireless network with signal reception depending on external influences). The third option, conjecture-based coding of expected actions, is only acceptable as a (tentative) workaround in cases where the other two solutions are unavailable. As will become clear in the following sections of the paper, our current understanding of some of the more complex natural processes underlying human actions, such as thought and perception, is insufficient to offer a solution in the form of practicable simulation (or emulation) algorithms. In those cases, conjecture-based instructions may be the only software-based solution that is available to close the loop.

Practically, we can say that the best available overall solution to investigate interaction between humans and artefacts based fully on software is most likely a combination of the three standalone options. It should *as much as possible* be based on true simulation software. Where simulation is not possible, emulation should be considered, and where emulation is not possible, conjecture-based instructions may offer a workaround. If, to reproduce particular actions or operations, even conjecture-based instructions are inconceivable, we have to conclude that closing the control loops fully based on software is not possible. For a virtual product testing approach based on this schedule of priorities, it will be possible to clearly point out which outcomes can be considered scientifically sound and which outcomes have to be interpreted with care.

2.2. Continuous and Discrete Control. In modelling and specifying control processes, a fundamental distinction is the one between continuous control and logical (discrete) control. Continuous control mechanisms are based on signals

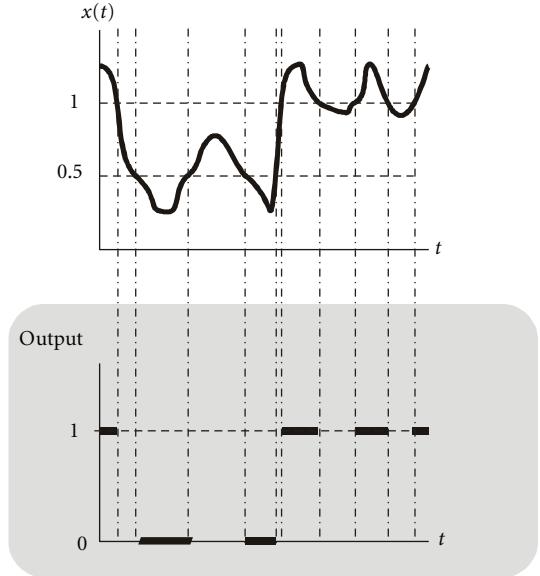


FIGURE 2: Discrete interpretation of continuous change of position.

directly corresponding to physical phenomena, for example, displacement as resulting from motion. The behaviour of such mechanisms is typically simulated based on constructs such as algebraic descriptions, block diagrams, and bond graphs, which the authors have evaluated in [5, 6]. For the paper here, we have deemed it sufficient to discuss the *application* of continuous control simulations to humans and artefacts. Generally, scientifically validated algorithms are available for continuous control so that it lends itself well to true simulation (e.g., [11]).

On the other hand, in logical or discrete control mechanisms logical operations are performed on information that is *interpreted* from physical effects that can be observed in signals (Figure 2). The logical operations are either *simulated* or *executed* based on constructs that disregard the physics background of the signals and that have not extensively been discussed in [5, 6]. In Section 5, the various representations for logical control are reviewed based on their representation potential and ease of use.

In the next section, the specific types of control in artefacts and humans are introduced and further elaborated.

3. Control Behaviours of Artefacts and Humans

3.1. Control of Artefacts. In artefacts, control is usually implemented to engage actuators based on input from sensors (e.g., [12]). Mechanisms for control in artefacts are subdivided into two main categories, namely, analogue control and digital control. These correspond to continuous and discrete control, respectively. Typically, in continuous control a control signal constantly applies corrections to a controlled signal so that the corrected output signal remains close to a set value.

A common control mechanism is the closed feedback loop where the control signal is derived from (a measurement

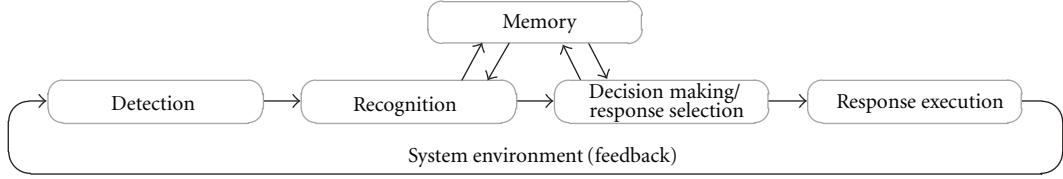


FIGURE 3: Generic stages of human control in interaction.

of) the output signal, and which is often based on the proportional/integral/derivative (PID) algorithm [13].

In the artefact itself, continuous control is usually realised by hardware components that process physical quantities (e.g., voltage or pressure) as signals. Most commonly analogue electronics is used, but continuous control can also be hydraulic or mechanical (e.g., centrifugal governors, [14]). Such mechanisms are typically simulated based on differential equations, which describe the physics variable carrying the control signal (e.g., electrical voltage and hydraulic pressure) or alternatively a continuously changing information flow derived from that variable [15]. On the other hand, embedded digital control mechanisms, such as microcontrollers and programmable logic devices (PLDs), are usually simulated as discrete-event systems performing logical operations on interpreted signals [16].

3.2. Control of Humans. Theories on human control cover a wide variety of responses to external stimuli, such as walking, looking, reaching, grasping, drawing, keyboarding, and speaking [17]. Human motor scientists have developed various approaches to model the control of interactions. In analogy with embedded control in artefacts, human control activates muscles (human actuators) based on input from perception (human sensors). The literature agrees that signals from the sense organs are processed by the central nervous system and alternately by the brain. This happens through a series of stages. Several interpretations have been proposed for this structuring.

For instance, Stelmach proposed a simplified decomposition scheme with four generic stages, namely, detection, recognition, response selection, and response execution, with recognition and response selection relying on memory [18]. These generic stages have a lot in common with other subdivisions, such as the ones proposed by Wickens and Hollands [19] and by Parasuraman et al. [20], who distinguish decision making or cognition as an additional stage before response selection. In Figure 3, we have brought together the common elements of the various subdivisions into stages. In fact, detection and recognition are considered preparations for the inputs of the actual control, memory stores knowledge on how to control, and decision making, response selection, and response execution represent the actual control. Involving conscious cognitive processes, decision making and response selection are typically considered “high-level” control, whereas response execution (i.e., the actual motion of limbs) is treated as “low-level” control [21]. Decision making in human motion control is not to be

confused with other high-level decision-making actions on a social level (e.g., [22]).

Interaction (or any sequence of multiple body movements) can be considered a succession of *key postures* with movements in between, where each key posture is the end posture of the movement before it and the start posture of the movement after it [23]. Decision making concerns the plan to assume a next end posture with a particular intention (e.g., *to grasp handle with right hand*), response selection concerns choosing the end posture in terms of joint angles and contact surfaces of interaction, and response execution concerns controlling the movement between the current posture and the next one.

Decision making is concerned with the *why* behind selecting a response. It can be modelled based on logic, as is commonly accepted in contemporary cognitive psychology [24]. Response selection has been considered a hierarchical structure of options (e.g., different grasping configurations for the hand) in which a choice is logically derived based on conditions determined by the task at hand [21, 25]. The resulting options (e.g., “precision grip” or “power grip”) may prescribe which surfaces of which body parts interact with other objects [26], but the actual joint configuration and whole-body posture depends on the position and orientation of the object relative to the human [27]. Response execution has been described as an eye-limb coordination exercise [28]. Findings over the past decennia indicate that the human brain specifies the input to response execution as movements based on positions, angles, velocities, and angular velocities rather than on forces and accelerations (e.g., [29]). Other researchers have successfully generalised and parameterised motion patterns that completely describe transitions between particular given ranges of start postures and end postures. These motion patterns have been called *invariants* [30]. A considerable advantage of invariants is that they effectively allow us to ignore the influence of eye-limb coordination in response-execution modelling.

As we just mentioned, modelling of decision making is typically done based on logic. Regarding the other stages of control, there is still much debate in the literature whether these are to be modelled as discrete (logic-based) or continuous. The *discrete-information processing theory* considers and models the human as a processor of information, comparable to a computer [18], and as a composition of receptors, effectors, and an intervening control system, with information processing concerned primarily with the operations of the control system.

The applications of *continuous* simulation models in the literature concentrate on response execution. Costello [31]

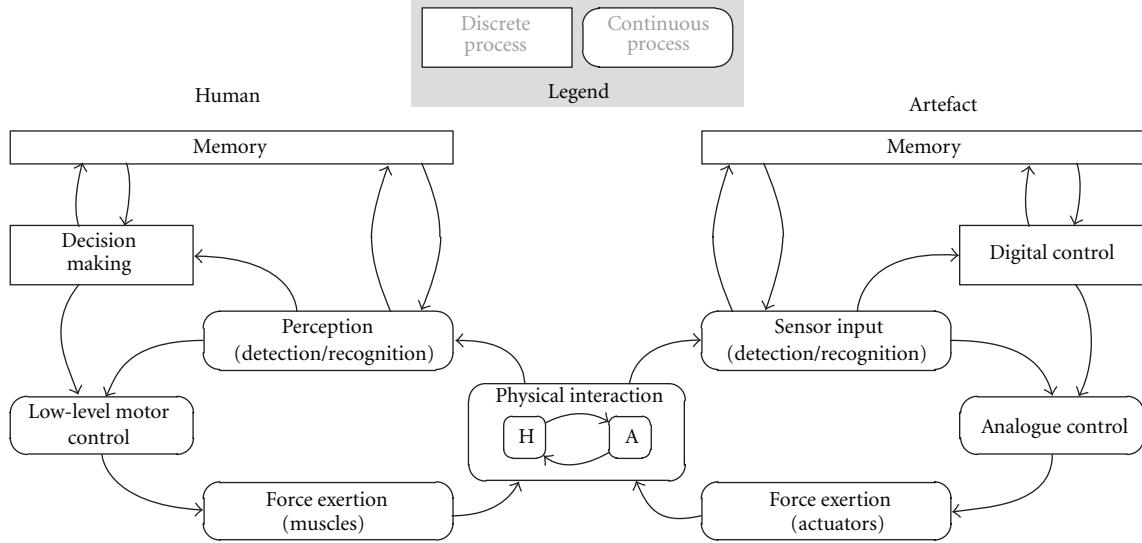


FIGURE 4: Simplified reasoning model of control loops in human-artefact interaction (further elaboration of Figure 1).

reserved continuous models for small corrections, while he considered large-scale movements to be better represented by discrete models. Sanders [32] proposed two stages of response execution, namely, (i) response programming, which suggests logical control, and which seems to correspond to Costello's large-scale movement planning, and (ii) motor adjustment, which is supposed to produce "instructed muscle tension", in other words, it specifies the force required. This also seems to be in agreement with findings that the velocity and position control signals that are input to response execution are generally considered to be discrete and pulsatile [33].

Findings from the literature (which does not appear to be decisive) suggest that computational modelling and specification control of human movements can be kept relatively simple by applying two principles. The first is to implement generalised and parameterised motion patterns. If these *invariants* are available, they can be used to circumvent simulation of eye-limb coordination, and thus disregard the role of perception in motor adjustments. The second principle is to model/specify response selection and execution as logical control, with the exception of the final translation of motion patterns to muscle forces, which is more appropriately modelled as continuous (e.g., PID) control.

3.3. Control Loops in Human-Artefact Interaction. As a wrap-up, Figure 4 shows how the various types and stages of control that the literature distinguishes in humans and artefacts are related, and how they come together in physical interactions. In most of the approaches reviewed in the next sections, the physics of interaction (central box in Figure 4), if included in simulation, is limited to a selection of mechanical behaviours, that is, rigid-body kinetics, deformations, and kinematics. What is also disregarded in most approaches, is how metabolic processes (digestion, breathing, blood flow, etc.) provide the energy needed for human cognition and

motion. We have not included these processes in the figure under the assumption that this omission is acceptable in forms of product use, where fatigue and exhaustion can be ignored, that is, outside applications such as sports and warfare.

4. Simulation of Continuous Control Behaviour of Artefacts and Humans

As was explained in Section 2, simulation of continuous control has already been discussed in [5, 6]. Since control behaviour is often unrelated to geometry and shape, the typical simulation constructs for continuous control behaviour are based on nongeometric representations such as block diagrams and bond graphs, for which various modelling and simulation packages are available [34, 35]. Several application examples can be found in [15] and many other publications. Matlab Simulink is often used to model block diagrams because many commercial packages for continuous simulation (e.g., finite element analysis and multibody dynamics) offer an interface with it [36, 37].

Continuous simulation of *human control behaviour* is typically applied to investigate response execution of movements. In the investigated literature three types of models have been used: (i) conventional signal-correcting control loops, (ii) posture frames that have been prerecorded or that are precalculated based on optimization algorithms, and (iii) artificial neural nets (ANNs).

Examples of behaviour simulated as a conventional control loop are eye-hand coordination in vehicle control (e.g., [38]) and interactive compensation of machine behaviour by McRuer [39]. In McRuer's loop, a machine reacts to human input and to external disturbances that are to be compensated by the human. In both examples, the physical transfer of information entered by the human into the machine—for example, by operating an interface element—is a step that is skipped. An example that includes physics was

proposed by Multon et al. [40], who combined simulation of low-level motion control with kinematics and rigid-body kinetics of the human arm.

In the case that series of posture frames are used to control human-motion simulations, these are typically obtained from motion-capture devices. For commercial human modelling and simulation packages that operate on geometric/anatomical models of the whole human body (e.g., LifeModeler [41, 42] and Anybody [43]), this is the most common form of input. Based on inverse dynamics these packages analyse the motions prescribed by the frames to compute muscle forces. This approach is inflexible because it only permits motions exactly as they have been captured from a specific human subject in a specific interaction, thus allowing open-loop simulations only.

Recent approaches have aimed to overcome this disadvantage by applying optimization algorithms to calculate incremental changes between arbitrary start and end postures. In “memory-based motion simulation” (MBMS) Park et al. [44] used a database with so-called root motions, which were captured from human subjects between particular pairs of start postures and end postures. To find motions between other pairs of postures, the most similar root motion was retrieved from the database and then adapted to match the new start and end posture by minimizing the deviation from the root motion. In the Santos virtual human, Yang et al. [45] implemented a different, so-called “multiobjective” optimization algorithm to calculate the most likely intermediate postures based on minimum potential energy, preference for “neutral postures”, and three other factors.

By considering the end position before computation these approaches effectively create a closed loop in low-level control without considering feedback based on perception. In fact they generate invariants (see Section 3.2) between arbitrary start and end postures rather than between fixed sets of postures from experimental data. The need for precalculation, however, implies two potential disadvantages: (i) during the simulated movement, intervening interactions with other bodies may occur that necessitate recomputation and (ii) precalculation of motion frames unevenly distributes the computational load over the simulation runtime, because the whole motion must be computed before the simulated movement can start—nevertheless, it has been claimed that Santos can perform simulations in real time [46]. Another problem that these approaches do not address is that the exact end posture must be specified beforehand (and not only the position of a landmark, such as the fingertip). The constraint-based technique proposed by Singh et al. [47] uses a conventional ergonomic manikin to compute whole-body postures given the position and orientation of one or more product surfaces to be gripped by the hands of a conventional ergonomic manikin. However, it does not compute transitional motions between different gripping situations and it does not support kinetics simulation.

The third group of approaches is based on ANNs, which form a particular class of block diagrams [34]. They can be considered discrete models of human body components, that is, of biological neural nets (however, usually, there is no one-to-one correspondence between the components in an ANN

and the components of a biological neural net.) ANNs differ from conventional simulation algorithms because of their ability to learn and self-organise, to generalise from training data, and to process information in other ways normally thought of as intelligent [48]. An example is the simulation of steering behaviour of airplane pilots that was performed by Martens [49]. In this simulation, no geometric/anatomical model of the human body was used and the actuation of muscles and the physical interaction between the human and the controls of the plane were not considered. Instead, the output control signals of the human were directly converted to positions of the control stick. Kim and Hemami [50] used an ANN to simulate motions of the head and torso. The ANN acted on the output of a “desired trajectory generator”, which represented cognitive decisions about movements of the head. This unit was not elaborated.

Reil and Husbands [51] used ANNs in simulations of human walking. Based on evolutionary principles a best-performing network was selected. The approach has been further developed for application with geometric/anatomical models of the whole human body and brought to the market as the commercial software package Endorphin (<http://www.naturalmotion.com/>) which has successfully been applied to generate human motion animations for the entertainment industry [52]. To prepare simulations of action sequences, typical “behaviours” selected from a library (e.g., “jump”, “stagger”, and “writhe”) are scheduled on a timeline together with events such as predefined occurrences of forces acting on the human body [53].

5. Logical Control in Simulations

5.1. Formal Representations. The most common logics-based representations of control mechanisms and processes are *finite automata* or *state machines*. They prescribe transitions between states of a system, which are triggered by specified input [54]. Control signals can be assigned to transitions or states as output [55]. Three categories of logics-based representations can be distinguished: (i) formal language based (using procedural logic in the form of text and declarative codes), (ii) algebraic/numeric (using matrices, Boolean algebra, and temporal logic), and (iii) graphical [56].

Graphical representations include informal graphical symbol constructs typically based on directed graphs [57] and formal, numerically processable symbolic constructs. We have concentrated on these below. Graphical representations such as the one in Figure 5 are generally easier to comprehend, since (i) hierarchy and parallelism in complex systems can be shown more clearly, (ii), graphics allow selective reading depending on the level of details needed, and (iii) the number of concepts to be held in short-term memory is smaller [58]. Graphical representations with multiple uses are state transition diagrams (STDs) [59], also known as event graphs [60], Markov models [61], Petri nets [62], and statecharts [63]. Additionally, a large number of “dialects” of varying significance have been developed to extend these representations. The most important dialects of Petri nets are stochastic, timed, and high-level Petri nets [64], and the most

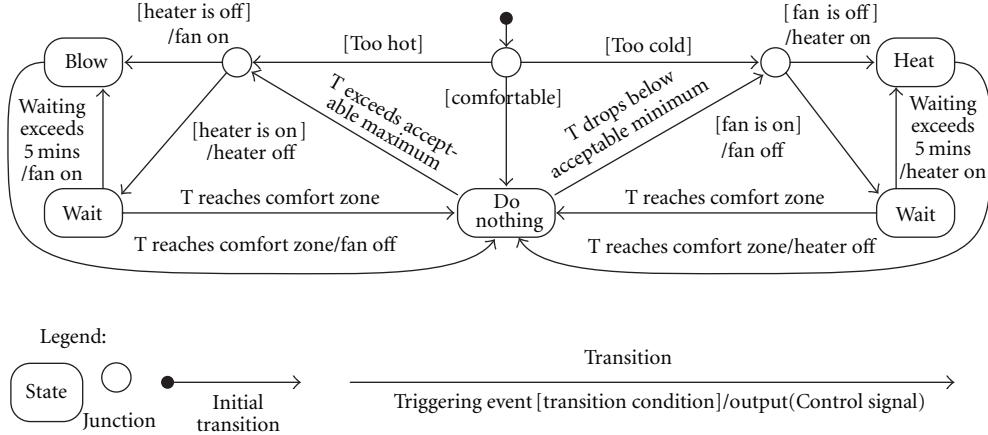


FIGURE 5: Example of a simple state machine: statechart of embedded control in a heating/cooling system.

important dialects of statecharts are stochastic and timed statecharts, as well as modecharts [65–67]. Various software packages exist for specification, modelling and simulation of STDs [68], Petri nets [69], and statecharts [63].

Considering its basic representation potential, the STD has become the most elementary representation for finite automata. It specifies transitions between global states of the whole control system and it can be converted to any other of the other, more “advanced” representations. Other conversions are also possible, that is, from Petri nets to statecharts [70, 71] and vice versa [72]. It should be noted that these conversions have been elaborated only for Petri nets and statecharts with particular characteristics.

The “advanced” representations have been developed to support (i) probabilistic transitions (Markov models, stochastic Petri nets, and stochastic statecharts), (ii) timing, that is, delayed transitions (timed Petri nets, timed statecharts, and modecharts), (iii) distributed or concurrent states (Petri nets and statecharts), and (iv) hierarchical decomposition (statecharts and high-level Petri nets).

Probabilistic transitions are needed to model nondeterministic systems. The capability to represent timing makes it possible to model countdown timers and latency in control loops. Distributed states, concurrency, and hierarchy are applied to avoid “state explosion”, that is, the need for a large number of states and transitions in STDs and Markov models, which makes these representations hard to work with [67]. Based on these considerations, it can be concluded that statecharts and Petri nets, which support distributed states, concurrency, and hierarchy (either by default or by using dialects), offer the highest representation potential.

5.2. Logical Control in Artefacts. In physical artefacts, digital circuits and embedded software are the typical discrete subsystems for which automata have been used as a representation. These subsystems receive input from sensor components and produce output that specifies the activity of actuator components [12]. In the development of products and systems, automata are often deployed as virtual prototypes of digital circuits or embedded software.

Letting automata (which may be linked to a simulation model) execute the instructions intended for the physical discrete subsystems is a form of emulation. After successful emulation, hardware designs and even fully functional (embedded) software can be created automatically from automata representations (e.g., [73]).

Specifically in simulations of manufacturing logistics, Petri nets have become the preferred virtual prototyping representation for control of machines and plants [67]. In a Petri net, distributed states are modelled with discrete tokens, which can be interpreted as processed units distributed over the plant (e.g., [74]). The popularity of Petri nets in industrial automation can also be attributed to the fact that the international standard for sequential function charts, which are used to design PLDs, is based on them [75].

Statecharts have become the prevalent representations in virtual prototyping of most other products and systems, for example, cars and airplanes [76]. Vahid and Givargis [16] elaborated on the use of statecharts in virtual prototyping of elevator control systems. Mosterman and O’Brien [77] and Pischinger et al. [78] used statecharts for closed-loop control over continuous simulations of power windows in cars, and fuel cells, respectively.

Other automata representations have also been applied in simulations of consumer products. For instance, Martell [79] and Bruno et al. [80] used STDs and statecharts, respectively, to emulate user interfaces of microwave ovens in HIL simulations, and Christensen [81] used a Petri net to emulate a linking device for networked audiovisual equipment.

5.3. Logical Control in Humans. Logical control appears in three types of human simulation models. The first category is that of models addressing only one of the stages depicted in Figure 3, in particular decision making and response selection. They cannot be directly used in interaction simulations but they can be used as simulation components. The second category addresses specific use processes on an abstract level. It does not involve models of the human body. The third, most advanced category is integrated into full geometric/anatomical human body, which means that,

potentially, in use-process simulations, even the physical part of the human interaction loop can be closed.

The decision-making models in the first category are known as “cognitive architectures” (CAs). Developed and empirically validated by psychologists, CAs (such as ACT-R and EPIC) are production-rule-based blueprints of the operational structure of cognition [82, 83]. In contrast to all the other models reviewed in this paper, only CAs explicitly include the workings of human *memory* in simulations. Although memory is not in the primary control loop (Figures 3 and 4), the time it takes to access it has to be regarded in simulations. Production rules for building CAs have been prepared for specific interaction tasks with specific products, especially for use processes that involve typical frequently occurring tasks, in particular driving cars [84], piloting aircraft [85], and interacting with computers [86]. For tasks related to the use of other products, new production rules need to be developed and validated. An important limitation of CA simulations is that they predict the *time* the human brain needs for processing perceptual input and for directing motor operations, but not the *reasoning* that determines which operation is performed in which situation. CA-based simulations start from a given *task decomposition*, that is, an idealised sequence of operations as it can be extracted from an instruction manual.

A response selection model belonging in the first category is the hierarchical structure in [25] that has been proposed in line with [21] to select grasping responses based on logical evaluation of shapes and dimensions of objects. Response execution models that allow us to disregard perception have been proposed based on invariants (i.e., generalised and parameterised motion patterns) [30]. Bullock and Grossberg [87] used invariants identified in [88] to describe human reaching with differential equations, for which logic-based algorithms provided parameter values. There are indications in the literature that invariants also apply to grasping [89]. However, due to the large number of possible variables and the redundancy of applicable grasping patterns, this issue needs further study [90].

The second category typically applies a radical approach to modelling of interaction control by bypassing both perception and human motor control. This can be achieved by directly modelling the effects of human control on the artefact with which the human interacts. This approach, which was also taken by [49] in Section 4, is particularly feasible if the artefact is a given specific product, for example, an airplane or a car. For instance, using an STD combined with evolutionary algorithms, Fogel and Moore [91] modelled response execution of pilots as motions of the steering components of the airplane rather than motions of the arms and hands. Liu and Salvucci [92] presented a model of human decision making based on a hidden Markov chain, which was used to simulate driving behaviour by computing accelerations and direction changes of cars. In Section 4, we found that similar shortcuts have been applied in continuous control simulations [39, 49].

The combined human models in the third category also incorporate various subsets of constructs and models that

have been reviewed in [5, 6], and they involve some logic to provide high-level control.

Perhaps the most well-known example in this category is the *Jack* manikin that was originally developed at the University of Pennsylvania [93, 94] and nowadays commercially marketed by Siemens PLM as *Vis Jack*. Jack is a semi-autonomous virtual human that makes decisions based on commands interactively entered by the user acting as human in the loop. The commands are conjectures about decision making formulated in near-natural language, for example, “walk around the room” [95]. They activate behavioural models of lower-level decision making that have been coded in a language-based (C++) finite-automaton representation called “parallel transition networks”. Lower-level control is based on stored sequences of animation frames. Additionally, Jack offers limited possibilities to simulate the physics of interaction with artefacts (kinematics and quasistatic force computations).

Carruth et al. [96] presented an ACT-R-enabled digital human model, ACT-R/DHM, based on Santos (see Section 4). Developing new production rules for CAs is a laborious endeavour, which may explain why the ACT-R/DHM researchers so far have only considered the use of one product, a vending machine, in their efforts to apply CAs to control of dynamical simulations with 3D human models [96]. So far, the most progress they made has been in vision simulation, particularly in predicting the time needed to visually recognise interaction features (e.g., buttons) in 3D static product models [97]. Based on the perceived features, the ACT-R cognition simulation activates consecutive responses to interact with the product (e.g., “reach for the button”), which Santos’s low-level control simulation algorithms convert to movements. As is common in CA simulations, the consecutive responses are read from a conjectured linear task decomposition. The control loop is closed by the vision simulation that feeds the progress of the task back to the cognition simulation. The completion of the response (or task) acts as a condition based on which the next response is selected.

So far, the project does not include behaviours manifested by artefacts, and it is limited to control of simple tasks. A disadvantage of using cognitive architectures is that new tasks must be included by adding modules to the production system, which is a labour-intensive endeavour involving empirical research with human subjects and procedural language-based programming of rules.

As an alternative to control exerted by simulated cognition, the original developers of Santos have proposed the consideration of *scenarios* to schedule high-level motor control in human-product interactions based on conjecture [98]. Actually, the *scenario* (also known as *use case*) appears to have become the prevalent umbrella concept for the specification of conjectured human actions, in particular on the level of decision making. As such, it has been widely applied in software development, for requirements engineering and also simulation, but so far, its application to forms of interaction other than between human and computer has largely been limited to the use of informal representations (e.g., storyboards) to support creative processes [99–101].

Being the prevailing alternative to cognitive simulations for decision making, the concept of scenarios is now further elaborated in the next section by (i) giving an account of the literature explaining the concept and (ii) reviewing its current applications to simulation of use processes.

6. Scenario-Based Control in Interaction Simulation

A scenario of use has been defined as one possible way for a human user to control his interaction with a given product in given surroundings. Its execution usually means going through a series of choices from subsequently available options. In the context of use of consumer products, Stanton and Baber [102] explained scenarios by referring to Newell and Simon's theory of human problem solving [103]. The viewpoint they adopted is that the goal of using a product is to solve a problem. To that end, the user moves through a decision tree from the initial state "problem unsolved" to the goal state "problem solved", selecting between available operations. Each junction in the tree has various paths representing state-transforming operations, of which one is selected. Each of the possible routes that connect junctions is a scenario of use, and the set of all possible scenarios forms a scenario tree [104].

This common interpretation has been criticised for two reasons. Firstly, if use actions fail, the scenario does not end in a goal state "problem solved". Therefore, "negative" scenarios should also be considered [105]. Secondly, the tree representation is known to have limitations in terms of flexibility of representation. Therefore, more general terms like "organised sets of scenarios" or "scenario networks" have been suggested to include other, more flexible arrangements of possible scenarios (e.g., [106]). Actually, if we consider the state machine in Figure 5 as a description of user operations for a manually operated heater/fan combination, it is an example of an organised set of scenarios. Unlike a tree, which is built up by divergent branching only, it also contains loops and convergent branching.

There have been efforts to use organised sets of scenarios to achieve logical control over concrete operation and/or simulation processes. These have been typically implemented as control models by using the representations discussed in Section 5. In software engineering, organised sets of scenarios are commonly used in requirements specification, verification and prototyping [106]. Since software prototyping is usually done by executing or emulating the program under development, while physical interaction with hardware (mouse, keyboard, etc.) is usually not investigated, additional simulations are not needed. The logical control representations typically used are statecharts and, to a lesser extent, Petri nets and STDs [106–109]. The dominance of statecharts in this application area can possibly be attributed to the fact that they have become a standard representation in the unified modelling language UML [110].

Outside the domain of software engineering, the majority of the scenario-based control approaches have been developed for computer animations in training, gaming, and en-

tertainment. In these approaches movements are mostly generated based on key framing, that is, on predefined frame sequences rather than on simulated physics [111]. A more advanced approach was implemented for training purposes in the "Iowa driving simulator" [112]. This simulator projected virtual humans driving around in virtual cars, and it performed physics simulation controlled by combined scenarios specified as statecharts. In contrast to the approach in software prototyping, these covered not only human decision making, but also the control by artefact subsystems.

There is no general consensus on whose actions scenarios should describe. In their strictest sense, scenarios in their application to the use of products as explained in [102] are restricted to conjecture about user actions. This interpretation is seconded by some approaches [111, 113]. Other approaches do not explicitly distinguish human control from artefact control in scenarios (e.g., [106]), they include actions by other humans (e.g., other traffic participants in the Iowa driving simulator), and/or they generate instructions for humans in the loop [114].

In the support of product design, the application of scenarios as a means to control simulations of use processes with inclusion of human motor control and physical interaction has been scarce. In the investigated literature, only two references to the application of scenarios in this area could be found. The first was in a side remark on possible implementation in the Santos virtual human (see Section 5.3). In the other, Honglun et al. [113] claimed to have implemented scenarios to simulations of human-product interaction with a geometric/anatomical model of the whole human body. Unfortunately, implementation details are missing, and no reports on further developments after 2007 could be found. The timeline that can be specified with Endorphin (Section 4) and the task decomposition in ACT-R/DHM (see Section 5.3) are actually linear scenarios, that is, scenarios with only one path. The drawback of linear scenarios is that if a task is not completed as expected, no alternative path can be selected and the simulation goes into an open loop or might even crash.

7. Our Approach: Scenario Bundles for High-Level Control of Interaction Simulation

Figure 6 shows the approach to simulation of human-artefact interaction that was recently developed at Delft University of Technology and has been elaborated in [7–9]. Scenario bundle-based interaction simulation (SBIS) offers designers the possibility to specify foreseen actions by the user (i.e., human decision making) using a *scenario bundle*, that is, based on a state machine representation. In the proof-of-concept implementation of SBIS, which is shown in the figure, we used statecharts to represent state machines using Matlab Simulink. The approach also provides for specification of embedded logical control in artefacts, and some analogue control to convert prescribed displacements and velocities to forces, as required by the forward dynamics-based multibody simulation software we used, that is, MSC Adams. The figure shows images from a simulation of the

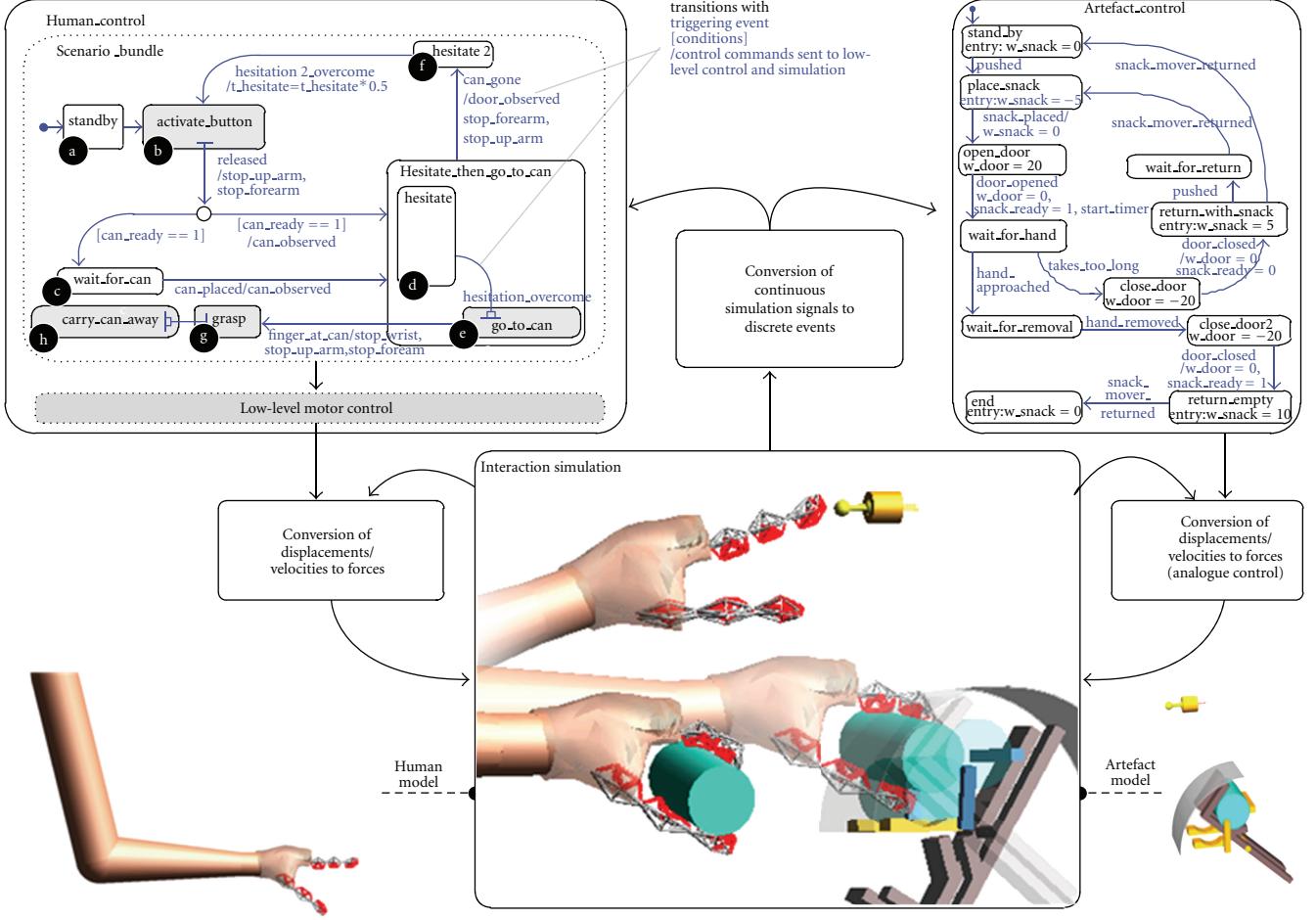


FIGURE 6: Simulation of human-artefact interaction based on scenario bundles.

use of a can dispenser. The scenario bundle holds multiple interaction sequences, such as *abdegh*, *abcdegh*, *abcfbcdegh*, and *abcdefbcdegh*, all of which we could use to control simulations.

In enabling simulation of interaction between human-body models and artefact models, the priority in SBIS has been to close the primary loops in Figure 4 by including logical control processes that the previously discussed other approaches have not addressed, or (to include human actions) have addressed by deploying a human in the loop. On the other hand, SBIS in its current form lacks validated algorithms for (i) kinematically determining the successive end postures of movements, (ii) simulating the low-level human motor control between those postures, and (iii) simulating time lags caused by cognitive processes. Instead, as a workaround the proof of concept relied on manually synthesised end postures that sufficed to perform the needed actions, such as pushing a button and grasping an object. The movements in between were programmed as robot-like successions of muscle-control commands that resulted in the next end posture without striving for realism. For some mental processes (hesitations), durations were included as delays of an arbitrary, conjectured amount of time expressed in seconds.

8. Discussion

From the viewpoint of application in human-artefact interaction simulation, control mechanisms appear both in products (artefacts) and in humans. Control mechanisms in *products* can be continuous and/or discrete. A simulation approach that supports comprehensive simulation of use processes should include this control. For modelling and simulation of continuous control in products, the industry typically uses block diagrams, which, therefore, can be considered the most favoured representation form. For modelling and emulation of discrete control, several representations are in use. Graphical representations are gaining popularity over text-based descriptions, because they are easier to use and to comprehend. Outside the area of manufacturing logistics, where Petri nets prevail, statecharts seem to be becoming the dominant graphical representation form in the industry.

Control in *humans* applies to interaction with virtually any product; therefore, its inclusion in simulations is even more essential than that of control embedded in products. Human control manifests itself chiefly in motor control, that is, control of muscles in order to move limbs and other body parts. Human-motor scientists have distinguished high-level

control, which relies on cognition, and low-level control, which is subconscious and relies on eye-limb coordination. In many simulation approaches with virtual humans, high-level control is not simulated but performed by a human in the loop.

The timing of the cognitive processes in high-level control can be simulated based on cognitive architectures such as ACT-R. However, for the *reasoning behind* decision making we could not identify any existing simulation approach. As a substitute, conjectured human decision making specified as a scenario can be used that is executed by software in the loop. So far, the use of scenario specifications to control use-process simulations has largely been restricted to the field of human-computer interaction, where continuous processes such as motor control and physical interaction are generally disregarded. Scenarios can be implemented using graphical representations, which allow designers to “play” with possible alternative unfoldings of use processes. Graphical logical representations may also allow designers to “bundle” scenarios into networks corresponding to multiple possible courses of use processes. One obvious set of scenarios is the one that can be obtained by formalizing the (draft) user manual of the product in question. This set of scenarios can be considered reasonably realistic, even if it cannot always be verified if real users act according to it. However, in product testing the most interesting scenarios are often those that correspond to unintended use. If no experimental data from human subjects is available, such scenarios can only be conceived through conjecture or by systematic variation of the actions according to the user manual (following alternative orders of operation, skipping actions, etc.). Common logical representations that are often used for scenarios, such as statecharts, have already successfully been applied for closed-loop control of (continuous) simulations of physics in artefacts.

Low-level control in existing virtual human simulation approaches is either based on prerecorded motion-capture frames or animation frames, which is an open-loop approach, or simulated. A constraint-based simulation approach has been proposed for detailed kinematical determination of the next end posture. To realistically simulate the movements to each next end posture (i.e., low-level motor control) without having to include eye-limb coordination in the computation, recent optimization-based approaches can be used—provided that there are no interventions between start and end posture.

In design support systems, geometric/anatomical models of the whole human body have become popular means to tune dimensions of products to humans. Most of these are conventionally limited to static posture investigations. In this paper, more advanced human-body models have passed in review that also incorporate behavioural simulation capabilities for the mechanics of physical interaction. Some of these models have intentionally been developed to offer use-process simulation as a form of design support. To wrap up the review of currently available approaches, we have subjected these human-body models to a closer inspection in order to assess the state of the art in integrating control and various physics behaviours in interaction

simulation. This outline will serve as a starting point for the conclusions.

The overview in Figure 7 brings together our approach of SBIS (as outlined in Section 7) with the most advanced simulation approaches for human-body models that were reviewed in the preceding part of this paper (Sections 4–6) as well as some conventional ergonomic human models that were reviewed by Feyen et al. in 2000 [115]. Two main groups of capabilities of these models turned out to be discriminating, that is, (i) the control-related functionality that is included and (ii) the range of physics phenomena that can be included in simulations of interaction.

(i) *Control-related functionality.* For *human decision making and response selection*, none of the approaches uses validated algorithms or true simulation models. ACT-R/DHM and Endorphin allow specification of linear scenarios. Only in SBIS, conjectured decision making and response selection allows variations conditionally depending on feedback. For *durations of mental processes*, only ACT-R/DHM uses validated simulation algorithms. SBIS allows specification of delays caused by mental processes, but these are currently not based on scientifically validated data. The *kinematical configuration of the next end posture* can only be computed based on validated algorithms with the constraint-based posture prediction approach. However, this approach is a “one trick pony” that does not offer any additional simulation functionality. The other approaches rely on conjecture or on a human in the loop for specification of the next end posture. *Low-level motor control* to generate motions from one posture to the next based on validated simulation algorithms has been implemented in MBMS, Santos and ACT-R/DHM (which uses Santos’s algorithms for this purpose). Embedded control in artefacts can be included only in SBIS.

(ii) *Range of physical interaction behaviours covered.* Apart from the constraint-based posture prediction approach, all of the approaches support simulations of the *kinematics of human-body motion* and most approaches also support kinematics of artefacts. Full simulation of *rigid-body kinetics* of humans and artefacts is offered by LifeModeler, Anybody and SBIS. Other approaches offer limited support for kinetics or none at all. Simulation of *deformations* is only offered in some form by SBIS, but its current implementation lacks stability [9]. None of the approaches support simulation of interactions involving physics phenomena outside the realm of (solid) mechanics, such as heat transfer or fluid dynamics. Of all the aspects considered, this is the only one that is not covered by any of the investigated simulation approaches. A possible explanation for this is that the simulation approaches discussed here appear to be human-centred, and humans do not possess any nonmechanical actuators that can be controlled. Artefacts, however, can be equipped with such actuators, and nonmechanical behaviours do play an important role in interactions with products such as hairdryers, solaria and water taps.

The approach of scenario bundle-based interaction simulation that we have introduced allows inclusion of all the other behaviours in some way, but there are three aspects of control for which a workaround is used that

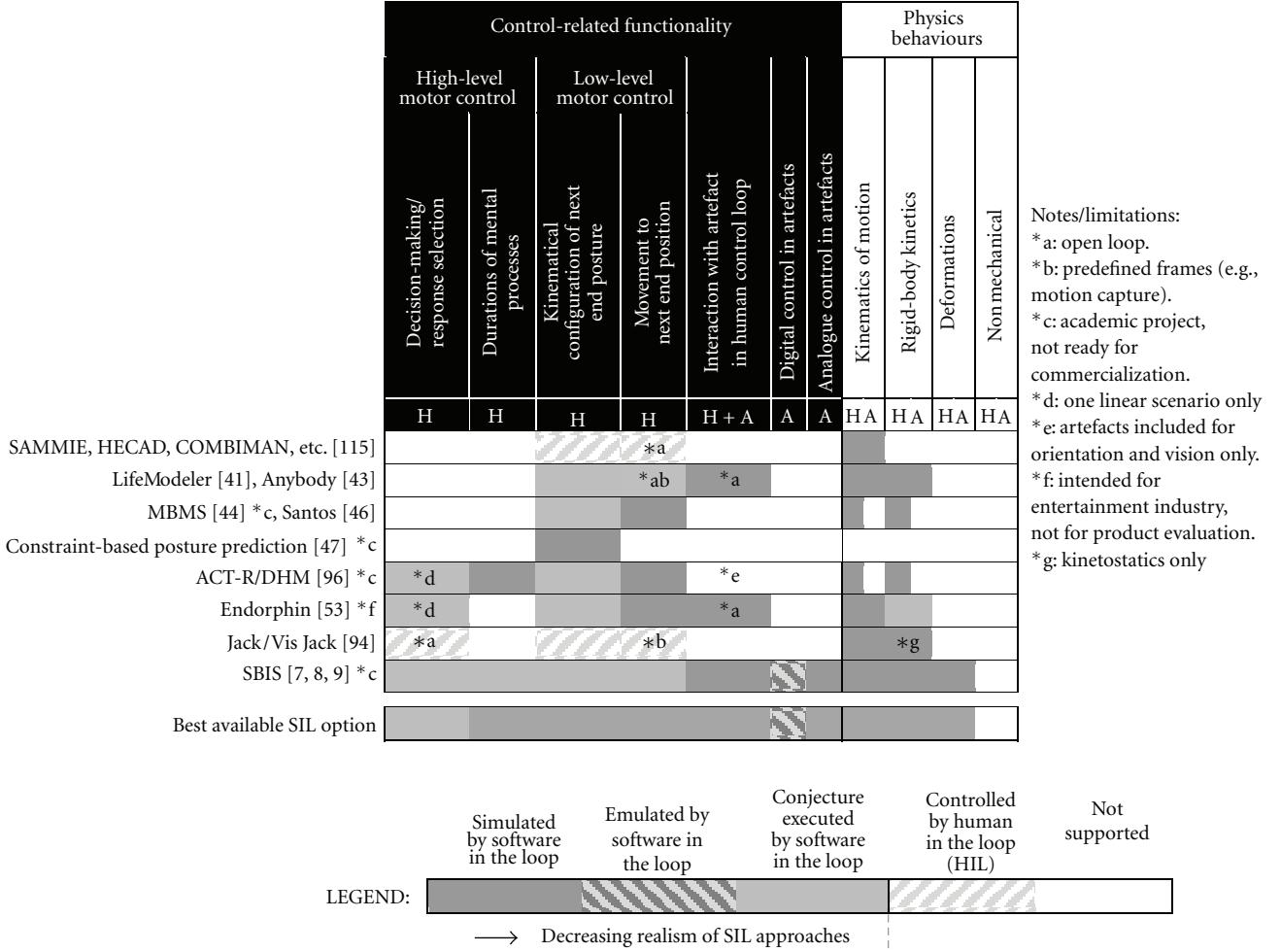


FIGURE 7: Review of full-body human simulations based on their capabilities.

can only produce a rough approximation for which other reviewed approaches already offer a better solution by relying on simulation algorithms that have been experimentally validated. These three concern the prediction of (i) durations of cognitive processes, (ii) kinematical configurations of each next end posture, and (iii) the movements needed to attain these postures. The other approaches, however, lack support for important other aspects of human-product interaction, such as the possibility of multiple interaction scenarios and the inclusion of artefact behaviours. The option to offer support for multiple conjectured interaction scenarios seems to be the best available option to include the logic behind human decision making in simulations, since validated simulation algorithms for human reasoning do not appear to become available in the near future.

9. Conclusions and Further Work

Geometric and anatomical models of the human body are frequently used by designers. The current models lack capabilities that allow these manikins to engage in interactions with virtual products as a software-in-the-loop

supplement to engineering simulations. In this paper, we have presented a simplified reasoning model describing the control loops involved in such interactions, and we have shown that all the ingredients needed to include all the involved behaviours (except for physical interactions outside the realm of solid mechanics) and to close these loops, can already be found in simulation approaches that have been proposed by academics or are even available as commercialised software packages. Our scenario bundle-based approach can be seen as a proof-of-concept that the loops can indeed be closed, in a way that resembles informal methods already in use by designers. However, to connect some links in the control chain it uses workarounds for which other available approaches offer better solutions in the form of experimentally validated simulation algorithms. Moreover, for the reasoning that underlies human decision making, no validated simulation approaches are likely to become available soon. One possible exception where a reasonable level of realism can be achieved using software-coded reasoning is using a set of decision-making scenarios that is programmed according to the user manual of the product. To summarize the current state of the art, we have to conclude that cur-

rently, no fully virtual approach can completely replace evaluations involving real humans interacting with physical prototypes or—through hardware interfaces—with virtual prototypes.

Nevertheless we believe that, if enhanced with knowledge from other approaches that that can extend the true simulation capabilities (see Figure 7), scenario bundles can offer an attractive addition to evaluations with real humans, because they currently offer the most advanced software-based means to close the “missing” link of human decision making in the interaction loop. We expect that the best opportunities lie in conceptual design, where simplified simulation models are often used and scientific rigour is not crucial. Simplified models of products allow simulations to run faster than real time, which allows simulations to be performed in batch to compare massive numbers of (parameterised) design variations. In addition, batch simulation can be applied to evaluate variations in human decision-making processes, for example, by applying stochastic variations to reaction times, or by randomizing selection of paths through the use process. Even if budgets would allow recruitment of human subjects for testing during conceptual design, such repetitive experiments would be hard to realise because of the total duration (due to the necessity to simulate everything in real time) and the risk of biasing influence that subjects can get from experience with preceding simulation runs.

As steps towards more realistic full software-in-the-loop simulation of human-artefact interaction, we propose to extend the scenario bundle-based approach by including (i) knowledge and algorithms from cognitive architectures to enable prediction of durations of cognitive processes, (ii) constraint-based algorithms to determine subsequent end postures, and (iii), optimization-based algorithms to compute movements between end postures. Apart from these simulation-related issues, we have not yet addressed the realization of full human-body models and the ability to generate anthropometric variations of users (different size, age, and gender). Incorporating this functionality should not be a bottleneck, since it is also available in existing software, most notably LifeModeler and Anybody. Aspects that may need attention in the more distant future are simulation of interaction processes relying on nonmechanical physics (or even multiphysics simulation in interaction), and, if at all feasible, development of validated models of human reasoning that allow true simulation of decision making. As a matter of course, all future additions to the system should be empirically validated (with the system as a whole) by using a human subject-based approach for product testing as a benchmark. This benchmark can be HIL simulation, but it can also testing with physical prototypes.

References

- [1] F. Ferrise, M. Bordegoni, and J. Lizaranzu, “Product design review application based on a vision-sound-haptic interface,” *Lecture Notes in Computer Science*, vol. 6306, pp. 169–178, 2010.
- [2] R. W. Pew and A. S. Mavor, *Human-System Integration in the System Development Process: A New Look*, The National Academic Press, Washington, DC, USA, 2007.
- [3] G. C. Burdea, “Haptics issues in virtual environments,” in *Proceedings of the 18th Computer Graphics International (CGI ’00)*, pp. 295–302, June 2000.
- [4] K. Tagawa, K. Hirota, and M. Hirose, “Impulse response deformation model: an approach to haptic interaction with dynamically deformable object,” in *Proceedings of the 14th Symposium on Haptics Interfaces for Virtual Environment and Teleoperator Systems*, pp. 209–215, Arlington, Va, USA, March 2006.
- [5] W. F. Van der Vegt, “A survey of artifact-simulation approaches from the perspective of application to use-processes of consumer durables,” in *Proceedings of the 6th International Symposium on Tools and Methods of Competitive Engineering (TMCE ’06)*, I. Horváth and Z. Rusák, Eds., vol. 2, pp. 617–632, Ljubljana, Slovenia, 2006.
- [6] W. F. Van Der Vegt and I. Horváth, “Including human behavior in product simulations for the investigation of use processes in conceptual design: a survey,” in *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC ’06)*, Philadelphia, Pa, USA, September 2006.
- [7] W. F. Van der Vegt and Z. Rusák, “Controlling simulations of human-artifact interaction with scenario bundles,” in *Proceedings of the Engineering Design in Integrated Product Development (EDIProD ’08)*, pp. 197–209, Jurata, Poland, 2008.
- [8] W. F. Van der Vegt, I. Horváth, and Z. Rusák, “Simulating the use of products: applying the nucleus paradigm to resource-integrated virtual interaction models,” in *Proceedings of the 7th International Symposium on Tools and Methods of Competitive Engineering (TMCE ’08)*, I. Horváth and Z. Rusák, Eds., vol. 1, pp. 591–605, Izmir, Turkey, 2008.
- [9] W. F. Van der Vegt, *Testing virtual use with scenarios*, Ph.D. thesis, Vereniging voor Studie en Studentenbelangen te Delft, Delft, The Netherlands, 2009.
- [10] E. G. Mallach, “Emulator architecture,” *Computer*, vol. 8, no. 8, pp. 24–32, 1975.
- [11] D. Assanis, Z. Filipi, S. Gravante et al., “Validation and use of Simulink integrated, high fidelity, engine-in-vehicle simulation of the international class VI truck,” *SAE Transactions*, vol. 109, no. 3, pp. 384–399, 2000.
- [12] J. M. Thompson and M. P. E. Heimdal, “Integrated development environment for prototyping safety critical systems,” in *Proceedings of the 10th IEEE International Workshop on Rapid System Prototyping (RSP ’99)*, pp. 172–177, Clearwater Beach, Fla, USA, June 1999.
- [13] W. J. I. Palm, *System Dynamics*, McGraw-Hill, New York, NY, USA, 2005.
- [14] S. Bennett, “Brief history of automatic control,” *IEEE Control Systems Magazine*, vol. 16, no. 3, pp. 17–25, 1996.
- [15] F. T. Brown, *Engineering System Dynamics—A Unified Graph-Centered Approach*, Marcel Dekker, New York, NY, USA, 2001.
- [16] F. Vahid and T. Givargis, *Embedded Systems Design—A Unified Hardware/Software Introduction*, John Wiley & Sons, New York, NY, USA, 2002.
- [17] D. A. Rosenbaum, *Human Motor Control*, Academic Press, San Diego, Calif, USA, 1991.
- [18] G. E. Stelmach, “Information-processing framework for understanding human motor behavior,” in *Human Motor*

Behavior—An Introduction, J. A. S. Kelso, Ed., pp. 63–91, Lawrence Erlbaum Associates, London, UK, 1982.

[19] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*, Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[20] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “A model for types and levels of human interaction with automation,” *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 30, no. 3, pp. 286–297, 2000.

[21] S. T. Grafton and A. F. D. C. Hamilton, “Evidence for a distributed hierarchy of action representation in the brain,” *Human Movement Science*, vol. 26, no. 4, pp. 590–616, 2007.

[22] M. J. Hicks, *Problem Solving and Decision Making*, Thomson Learning, London, UK, 2004.

[23] C. L. Elsinger and D. A. Rosenbaum, “End posture selection in manual positioning: evidence for feedforward modeling based on a movement choice method,” *Experimental Brain Research*, vol. 152, no. 4, pp. 499–509, 2003.

[24] M. W. Eysenck and M. T. Keane, *Cognitive Psychology*, Psychology Press, Hove, UK, 4th edition, 2000.

[25] M. R. Cutkosky and R. D. Howe, “Human grasp choice and robotic grasp analysis,” in *Dextrous Robot Hands*, S. T. Venkataraman and T. Iberall, Eds., pp. 5–31, Springer, New York, NY, USA, 1990.

[26] J. R. Napier, “The prehensile movements of the human hand,” *The Journal of Bone and Joint Surgery*, vol. 38, no. 4, pp. 902–913, 1956.

[27] X. Wang, “Three-dimensional kinematic analysis of influence of hand orientation and joint limits on the control of arm postures and movements,” *Biological Cybernetics*, vol. 80, no. 6, pp. 449–463, 1999.

[28] D. Regan, “Perceptual motor skills and human motion analysis,” in *Handbook of Human Factors and Ergonomics*, G. Salvendy, Ed., pp. 174–218, John Wiley & Sons, New York, NY, USA, 1997.

[29] W. Abend, E. Bizzì, and P. Morasso, “Human arm trajectory formation,” *Brain*, vol. 105, no. 2, pp. 331–348, 1982.

[30] C. G. Atkeson and J. M. Hollerbach, “Kinematic features of untreated vertical arm movements,” *Journal of Neuroscience*, vol. 5, no. 9, pp. 2318–2330, 1985.

[31] R. G. Costello, “The surge model of the well-trained human operator in simple manual control,” *IEEE Transactions on Man-Machine Systems*, vol. 9, no. 1, pp. 2–9, 1968.

[32] A. F. Sanders, “Stage analysis of reaction processes,” in *Tutorials in Motor Behavior*, G. E. Stelmach and J. Requin, Eds., pp. 331–354, North-Holland, Amsterdam, The Netherlands, 1980.

[33] P. K. Mutha and R. L. Sainburg, “Control of velocity and position in single joint movements,” *Human Movement Science*, vol. 26, no. 6, pp. 808–823, 2007.

[34] N. M. Karayannidis, *Advanced System Modelling and Simulation with Block Diagram Languages*, CRC Press, Boca Raton, Fla, USA, 1995.

[35] J. M. D. Filippo, M. Delgado, C. Brie, and H. M. Paynter, “A survey of bond graphs : theory, applications and programs,” *Journal of the Franklin Institute*, vol. 328, no. 5-6, pp. 565–606, 1991.

[36] H. Van der Auweraer, “Frontloading design engineering through virtual prototyping and virtual reality: industrial applications,” in *Proceedings of the 7th International Symposium on Tools and Methods of Competitive Engineering (TMCE '08)*, vol. 1, pp. 39–52, Izmir, Turkey, 2008.

[37] H. Van der Auweraer, “Virtual engineering at work: the challenges for designing intelligent products,” in *Proceedings of the 8th International Symposium on Tools and Methods of Competitive Engineering (TMCE '10)*, vol. 1, pp. 3–18, Ancona, Italy, 2010.

[38] M. Alexík, “Modelling and identification of eye-hand dynamics,” *Simulation Practice and Theory*, vol. 8, no. 1-2, pp. 25–38, 2000.

[39] D. McRuer, “Human dynamics in man-machine systems,” *Automatica*, vol. 16, no. 3, pp. 237–253, 1980.

[40] F. Multon, J. L. Nougaret, B. Arnaldi, G. Hegron, and L. Millet, “Software system to carry-out virtual experiments on human motion,” in *Proceedings of the 1999 Computer Animation, CA'99*, pp. 16–23, May 1999.

[41] A. Veloso, G. Esteves, S. Silva, C. Ferreira, and F. Brandão, “Biomechanics modeling of human musculoskeletal system using adams multibody dynamics package,” in *Proceedings of the 24th IASTED International Conference on Biomedical Engineering*, pp. 401–407, Innsbruck, Austria, February 2006.

[42] I. C. Kenny, E. S. Wallace, S. R. Otto, and D. Brown, “Validation of a full-body computer simulation model for the golf drive,” in *Proceedings of the Annual Conference of the Exercise and Sport Sciences Association of Ireland*, Munich, Germany, 2005.

[43] J. Rasmussen, M. Damsgaard, E. Surma, S. T. Christensen, M. De Zee, and V. Vonderak, “Anybody—a software system for ergonomic optimization,” in *Proceedings of the 5th World Congress on Structural and Multidisciplinary Optimization*, Venice, Italy, 2003.

[44] W. Park, D. B. Chaffin, B. J. Martin, and J. Yoon, “Memory-based human motion simulation for computer-aided ergonomic design,” *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 38, no. 3, pp. 513–527, 2008.

[45] J. Yang, S. Rahmatalla, T. Marler, K. Abdel-Malek, and C. Harrison, “Validation of predicted posture for the virtual human Santos™,” *Lecture Notes in Computer Science*, vol. 4561, pp. 500–510, 2007.

[46] K. Abdel-Malek, Y. Yang, R. T. Marler, X. Zhou, A. Patrick, and J. Arora, “Towards a new generation of virtual humans,” *International Journal of Human Factors Modelling and Simulation*, vol. 1, no. 1, pp. 2–39, 2006.

[47] B. Singh, B. Hicks, A. J. Medland, G. Mullineux, and J. M. Molenbroek, “A constraint based human model for simulating and predicting postures,” in *Proceedings of the 8th International Symposium on Tools and Methods of Competitive Engineering (TMCE '10)*, vol. 1, pp. 221–229, Ancona, Italy, 2010.

[48] K. Gurney, *An Introduction to Neural Networks*, UCL Press, London, UK, 1997.

[49] D. Martens, “Neural networks as a tool for the assessment of human pilot behaviour in wind shear,” *Aerospace Science and Technology*, vol. 3, no. 1, pp. 39–48, 1999.

[50] J. Kim and H. Hemami, “Coordinated three-dimensional motion of the head and torso by dynamic neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 28, no. 5, pp. 653–666, 1998.

[51] T. Reil and P. Husbands, “Evolution of central pattern generators for bipedal walking in a real-time physics environment,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 159–168, 2002.

[52] L. D. Paulson, “Researchers automate the digital animation process,” *Computer*, vol. 36, no. 4, p. 24, 2003.

[53] *User Guide: Endorphin™ Dynamic Motion Synthesis*, NaturalMotion, San Francisco, Calif, USA, 2006.

- [54] B. Khoussianov and A. Nerode, *Automata Theory and Its Applications*, Birkhäuser, Boston, Mass, USA, 2001.
- [55] D. T. Lee, “Evaluating real-time software specification languages,” *Computer Standards and Interfaces*, vol. 24, no. 5, pp. 395–409, 2002.
- [56] B. Hrúz and M. C. Zhou, *Modeling and Control of Discrete-Event Dynamic Systems*, Springer, London, UK, 2007.
- [57] C. H. E. Phillips, “Review of graphical notations for specifying direct manipulation interfaces,” *Interacting with Computers*, vol. 6, no. 4, pp. 411–431, 1994.
- [58] T. H. Tse and L. Pong, “Examination of requirements specification languages,” *Computer Journal*, vol. 34, no. 2, pp. 143–152, 1991.
- [59] A. Gill, *Introduction to the Theory of Finite-State Machines*, McGraw-Hill, New York, NY, USA, 1962.
- [60] L. Schruben, “Simulation modeling with event graphs,” *Communications of the ACM*, vol. 26, no. 11, pp. 957–963, 1983.
- [61] I. L. MacDonald and W. Zucchini, *Hidden Markov and Other Models for Discrete-Valued Time Series*, Chapman & Hall, London, UK, 1997.
- [62] C. A. Petri, “Fundamentals of a theory of asynchronous information flow,” in *Proceedings of the International Federation for Information Processing Congress (IFIP '62)*, pp. 386–390, Amsterdam, The Netherlands, 1962.
- [63] D. Harel, “Statecharts: a visual formalism for complex systems,” *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, 1987.
- [64] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri Nets*, Springer, Berlin, Germany, 2005.
- [65] C. R. L. Francès, E. Da Luz Oliveira, J. C. W. A. Costa et al., “Performance evaluation based on system modeling using Statecharts extensions,” *Simulation Modelling Practice and Theory*, vol. 13, no. 7, pp. 584–618, 2005.
- [66] J. Qian and B. Xu, “Model checking for timed statecharts,” *Lecture Notes in Computer Science*, vol. 3731, pp. 261–274, 2005.
- [67] A. M. K. Cheng, *Real-Time Systems*, Wiley-Interscience, Hoboken, NJ, USA, 2002.
- [68] D. V. Koznov, M. Kartashev, R. Gagarsky, V. Zvereva, and A. Barsov, “Roundtrip engineering of reactive systems,” in *Proceedings of the International Symposium on Leveraging Applications of formal methods (ISoLA '04)*, pp. 343–347, Paphous, Cyprus, 2004.
- [69] K. H. Mortensen, H. Rölke, F. Heitmann, and D. Moldt, “Petri nets world: online services for the international Petri nets community,” Universität Hamburg, 2005, <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>.
- [70] M. K. Schnabel, G. M. Nenninger, and V. G. Krebs, “Konvertierung sicherer Petri-Netze in Statecharts—conversion of safe Petri nets into statecharts,” *Automatisierungstechnik*, vol. 47, no. 12, pp. 571–580, 1999.
- [71] R. Eshuis, “Statecharting Petri nets,” Eindhoven University of Technology BETA Working Paper Series, 2005.
- [72] S. Bernardi, S. Donatelli, and J. Merseguer, “From UML sequence diagrams and statecharts to analysable petri net models,” in *Proceedings of the 3rd International Workshop on Software and Performance (WOSP '02)*, pp. 35–45, Rome, Italy, July 2002.
- [73] A. V. Yakovlev, A. M. Koelmans, A. Semenov, and D. J. Kin-niment, “Modelling, analysis and synthesis of asynchronous control circuits using Petri nets,” *Integration*, vol. 21, no. 3, pp. 143–170, 1996.
- [74] S. Julia and R. Valette, “Real time scheduling of batch systems,” *Simulation Practice and Theory*, vol. 8, no. 5, pp. 307–319, 2000.
- [75] P. Orth, A. Bollig, and D. Abel, “Rapid prototyping of sequential controllers with Petri nets,” in *Proceedings of the International Federation of Automatic Control World Congress*, Prague, Czech, 2005.
- [76] M. Otter, M. Remelhe, S. Engell, and P. Mosterman, “Hybrid models of physical systems and discrete controllers,” *Automatisierungstechnik*, vol. 48, no. 9/2000, p. 426, 2000.
- [77] P. Mosterman and E. O'Brien, “Hybrid dynamic systems in an industry design application,” in *Soft Computing in Industrial Applications: Recent and Emerging Methods and Techniques*, vol. 39, pp. 1–16, Springer, New York, NY, USA, 2007.
- [78] S. Pischinger, C. Schönfelder, and J. Ogrzewalla, “Analysis of dynamic requirements for fuel cell systems for vehicle applications,” *Journal of Power Sources*, vol. 154, no. 2, pp. 420–427, 2006.
- [79] A. Martel, “Application of ergonomics and consumer feedback to product design at Whirlpool,” in *Human Factors in Consumer Products*, N. A. Stanton and M. S. Young, Eds., pp. 75–90, Taylor & Francis, London, UK, 1998.
- [80] F. Bruno, F. Cosco, A. Angelica, and M. Muzzupappa, “Mixed prototyping for products usability evaluation,” in *Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC '10)*, pp. 1–10, Montreal, Canada, 2010.
- [81] S. Christensen, J. B. Jorgensen, and K. Halskov Madsen, “Design as interaction with computer based materials,” in *Proceedings of the 2nd Biannual Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '97)*, pp. 65–71, August 1997.
- [82] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, “An integrated theory of the mind,” *Psychological Review*, vol. 111, no. 4, pp. 1036–1060, 2004.
- [83] D. E. Kieras, S. D. Wood, and D. E. Meyer, “Predictive engineering models based on the EPIC architecture,” *ACM Transactions on Human-Computer Interaction*, vol. 4, no. 3, pp. 230–275, 1997.
- [84] D. D. Salvucci, “Modeling driver behavior in a cognitive architecture,” *Human Factors*, vol. 48, no. 2, pp. 362–380, 2006.
- [85] A. Lüdtke and J. P. Osterloh, “Simulating perceptive processes of pilots to support system design,” *Lecture Notes in Computer Science*, vol. 5726, no. 1, pp. 471–484, 2009.
- [86] M. Byrne, “Cognitive architectures in HCI: present work and future directions,” in *Proceedings of the International Conference on Human-Computer Interaction (HCI '05)*, Las Vegas, Nev, USA, 2005.
- [87] D. Bullock and S. Grossberg, “Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation,” *Psychological Review*, vol. 95, no. 1, pp. 49–90, 1988.
- [88] P. Morasso, “Spatial control of arm movements,” *Experimental Brain Research*, vol. 42, no. 2, pp. 223–227, 1981.
- [89] Y. Paulignan, V. G. Frak, I. Toni, and M. Jeannerod, “Influence of object position and size on human prehension movements,” *Experimental Brain Research*, vol. 114, no. 2, pp. 226–234, 1997.
- [90] J. Friedman and T. Flash, “Task-dependent selection of grasp kinematics and stiffness in human object manipulation,” *Cortex*, vol. 43, no. 3, pp. 444–460, 2007.

[91] L. J. Fogel and R. A. Moore, *Modeling the Human Operator with Finite-State Machines*, (NASA CR-1112), National Aeronautics and Space Administration, Washington, DC, USA, 1968.

[92] A. Liu and D. D. Salvucci, "Modeling and prediction of human driver behavior," in *Proceedings of the 9th International Conference on Human-Computer Interaction*, pp. 1479–1483, Los Angeles, Calif, USA, 2001.

[93] N. I. Badler, C. B. Phillips, and B. W. Webber, *Simulating Humans—Computer Graphic Animations and Control*, Oxford University Press, New York, NY, USA, 1993.

[94] N. I. Badler, M. S. Palmer, and R. Bindiganavale, "Animation control for real-time virtual humans," *Communications of the ACM*, vol. 42, no. 8, pp. 65–73, 1999.

[95] N. I. Badler, R. Bindiganavale, J. M. Allbeck, W. Schuler, L. Zhao, and M. S. Palmer, "Parameterized action representation for virtual human agents," in *Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds., pp. 256–284, MIT Press, Cambridge, Mass, USA, 2000.

[96] D. W. Carruth, M. D. Thomas, B. Robbins, and A. Morais, "Integrating perception, cognition and action for digital human modeling," *Lecture Notes in Computer Science*, vol. 4561, pp. 333–342, 2007.

[97] B. Robbins, D. Carruth, and A. Morais, "Bridging the gap between HCI and DHM: the modeling of spatial awareness within a cognitive architecture," *Lecture Notes in Computer Science*, vol. 5620, pp. 295–304, 2009.

[98] K. Abdel-Malek, J. Yang, J. H. Kim et al., "Development of the virtual-human Santos™," *Lecture Notes in Computer Science*, vol. 4561, pp. 490–499, 2007.

[99] K. Welker, E. B. N. Sanders, and J. S. Couch, "Design scenarios—to understand the user," *Innovation*, vol. 3, pp. 24–27, 1997.

[100] J. A. Landay and B. A. Myers, "Sketching storyboards to illustrate interface behaviors," in *Proceedings of the 1996 Conference on Human Factors in Computing Systems, CHI 96*, pp. 193–194, April 1996.

[101] A. Kankainen, "UCPCD: user-centered product concept design," in *Proceedings of the ACM Conference on Designing for User Experiences (DUX '03)*, pp. 1–13, San Francisco, Calif, USA, June 2003.

[102] N. A. Stanton and C. Baber, "A systems analysis of consumer products," in *Human Factors in Consumer Products*, N. A. Stanton, Ed., pp. 75–90, Taylor & Francis, London, UK, 1998.

[103] A. Newell and H. A. Simon, "Simulation of human thought," in *Computer Simulation of Human Behavior*, J. M. Dutton and W. H. Starbuck, Eds., John Wiley & Sons, New York, NY, USA, 1971.

[104] P. Hsia, J. Samuel, J. Gao, D. Kung, Y. Toyoshima, and C. Chen, "Formal approach to scenario analysis," *IEEE Software*, vol. 11, no. 2, pp. 33–41, 1994.

[105] S. Uchitel, J. Kramer, and J. Magee, "Incremental elaboration of scenario-based specifications and behavior models using implied scenarios," *ACM Transactions on Software Engineering and Methodology*, vol. 13, no. 1, pp. 37–85, 2004.

[106] M. Glinz, "An integrated formal model of scenarios based on statecharts," *Lecture Notes in Computer Science*, vol. 989, pp. 254–271, 1995.

[107] T. A. Alspaugh, D. J. Richardson, and T. A. Standish, "Scenarios, state machines and purpose-driven testing," in *Proceedings of the 4th International Workshop on Scenarios and State Machines: Models, Algorithms and Tools (SCESM '05)*, New York, NY, USA, May 2005.

[108] D. Amyot and A. Eberlein, "An evaluation of scenario notations and construction approaches for telecommunication systems development," *Telecommunication Systems*, vol. 24, no. 1, pp. 61–94, 2003.

[109] M. Elkoutbi, I. Khriss, and R. K. Keller, "Generating user interface prototypes from scenarios," *Proceedings of the IEEE International Conference on Requirements Engineering*, pp. 150–158, 1999.

[110] I. Khriss, M. Elkoutbi, and R. K. Keller, "Automating the synthesis of UML StateChart diagrams from multiple collaboration diagrams," *Unified Modeling Language*, vol. 1618, pp. 132–147, 1999.

[111] K. Perlin and A. Goldberg, "Improv: a system for scripting interactive actors in virtual worlds," in *Proceedings of the 1996 Computer Graphics Conference, SIGGRAPH*, pp. 205–216, August 1996.

[112] J. Cremer, J. Kearney, and Y. Papelis, "HCSM: a framework for behavior and scenario control in virtual environments," *ACM Transactions on Modeling and Computer Simulation*, vol. 5, no. 3, pp. 242–267, 1995.

[113] H. Hou, S. Sun, and Y. Pan, "Research on virtual human in ergonomic simulation," *Computers and Industrial Engineering*, vol. 53, no. 2, pp. 350–356, 2007.

[114] M. Tideman, M. C. van der Voort, and F. J. A. M. van Houten, "A new product design method based on virtual reality, gaming and scenarios," *International Journal on Interactive Design and Manufacturing*, vol. 2, no. 4, pp. 195–205, 2008.

[115] R. Feyen, Y. Liu, D. Chaffin, G. Jimmerson, and B. Joseph, "Computer-aided ergonomics: a case study of incorporating ergonomics analyses into workplace design," *Applied Ergonomics*, vol. 31, no. 3, pp. 291–300, 2000.