Chapter 14
LINDSAY Virtual Human: Multi-scale, Agent-based, and Interactive

C. Jacob, S. von Mammen, T. Davison, A. Sarraf-Shirazi, V. Sarpe, A. Esmaeili, D. Phillips, I. Yazdanbod, S. Novakowski, S. Steil, C. Gingras, H. Jamniczky, B. Hallgrimsson, and B. Wright

Abstract. We are developing *LINDSAY Virtual Human*, a 3-dimensional, interactive computer model of male and female anatomy and physiology. *LINDSAY* is designed to be used for medical education. One key characteristic of *LINDSAY* is the integration of computational models across a range of spatial and temporal scales. We simulate physiological processes in an integrative fashion: from the body level to the level of organs, tissues, cells, and sub-cellular structures. For use in the classroom, we have built *LINDSAY Presenter*, a 3D slide-based visualization and exploration environment that presents different scenarios within the simulated human body. We are developing *LINDSAY Composer* to create complex scenes for demonstration, exploration and investigation of physiological scenarios. At *LINDSAY Composer*'s core is a graphical programming environment, which facilitates the composition of complex, interactive educational modules around the human body.

14.1 Motivation

Health care systems all over the world are struggling to provide affordable and comprehensive care. Consequently, the need for excellent education and training of medical staff and doctors is more important than ever. At the same time, emerging digital technologies render it possible to present complex contents faster and better to a wide range of audiences. Within this educational context, we are developing *LINDSAY Virtual Human*, a project

C. Jacob · S. von Mammen · T. Davison · A. Sarraf-Shirazi · V. Sarpe · A. Esmaeili · D. Phillips · I. Yazdanbod · S. Novakowski · S. Steil · C. Gingras · H. Jamniczky · B. Hallgrimsson · B. Wright
University of Calgary, Calgary, Alberta, Canada
e-mail: cjacob@ucalgary.ca

J. Kołodziej et al. (Eds.): Advances in Intelligent Modelling and Simulation, SCI 422, pp. 327–349.
springerlink.com

at the intersection of Computer Science, Education, and Medicine. *LINDSAY* provides a collection of computational tools for research, presentation and learning in the context of human anatomy and physiology. *LINDSAY* also serves as a testbed for interactive computer graphics, touch-based user interfaces, multi-scale modelling, and scalable computing solutions to run simulations seamlessly on a diverse range of computing platforms: from web-based systems and desktop computers to laptops and mobile devices.

14.1.1 Starting with Virtual Anatomy

As a starting point for this project, we attended and analyzed lectures on human anatomy given by senior instructors from the Faculty of Medicine at the University of Calgary. One of our conclusions was that a virtual 3-dimensional human body in combination with the ability to easily navigate and label the displayed contents could greatly facilitate and improve the learning experience for students. Consequently, such an application for interactive 3D content presentation became the first milestone of our *LINDSAY* project.

14.1.2 Bringing Virtual Physiology to Life

LINDSAY Presenter supports the visualization and exploration of static contents on human anatomical structures. Integrating human physiological processes into an anatomy model is challenging. Highly dynamic processes—resulting from a densely connected network of interacting components—have to be illustrated over time and within 3-dimensional spaces. We did not want to resort to mere animations of such contents [3], as this would not allow for a wide range of interactive exploration of the underlying networked units and their interactions. Furthermore, we want users of *LINDSAY* to be able to navigate to any part of the virtual human body and, even more importantly, be able to quickly zoom to physiological processes at (almost) any scale. More precisely, a user should be able to experience human anatomy on the whole-body level, on the level of organs, tissues, and cells, down to sub-cellular components. This requires an integration of mathematical and computational models of the various processes across multiple scales in space and time. *LINDSAY Composer* is our first prototype of such a multi-scale programming, composition, and visualization toolset. It allows us to explore and present physiological processes within a virtual 3D female and male anatomy model.

14.2 Related Work

Before we expand on our approach to building a virtual human, we give a brief overview of virtual anatomy, the use of anatomy atlas databases, and the use of component-based frameworks for building dynamic software environments.

14.2.1 Replicating Human Anatomy and Physiology

For more than two decades, scientists have been exploring ways to enhance medical research and education through computer-based renderings of human anatomy and physiology. The American National Library of Medicine started as early as 1989 with the composition of a comprehensive imagery database of human physiology, also referred to as the Visible Human [4]. Since then, these data sets have inspired a large number of virtual anatomy projects for research, patient consultation and education [26].

14.2.2 Virtual Human Anatomy and Physiology

In the context of education, atlases have been composed that promote the exploration of detailed anatomical terminology in a proper visual context [21]. Some systems have been further extended to incorporate data about actual biomedical processes in the human body, for instance processes of gene control [27]. One of the projects at the forefront of modeling physiology by integrating processes across scales is the Physiome project [14], which takes signalling and metabolic pathways into account to connect protein models with cell simulations. The objective is to build tissues and organs through this bottom-up approach. Such systems can be modelled and simulated by means of traditional mathematical methodologies or as large sets of self-organizing bio-agents, or *swarm* systems [8, 15, 34]. The latter, agent-based approach forms the foundation for our computational models to implement physiological processes across multiple scales.

14.2.3 Components as Dynamic Building Blocks

In addition to multi-scale modelling, we rely on a component-based architecture for a framework to generate and deliver anatomical and physiological contents. The required diversity of data types and computational processes for an integrative simulation and presentation tool can be addressed by a component-based software architecture [12]. More specifically, a component can be broadly defined as a unit of independent deployment, which has a persistent state [18]. The design of component-based software architectures has advantages in regard to various application domains. Frameworks for (human) collaborative work can be implemented by brokering groupware components [29]. The coordinated execution of heterogeneous components works for organizing human collaboration as well as complex code bases that comprise large sets of interoperable, reusable software components. For example, a component-based augmented reality framework could manage elements and modules for user interfaces, tracking or object modelling [6].

Computer games face a similar challenge of integrating vast numbers of software components, whether related to contents, providing networking infrastructure or user interfaces [28]. Component-based frameworks have been investigated and applied in the context of Massively Multiplayer Online games [1]. For practical reasons, multi-facetted game units are defined by aggregating distinct software components rather than by using established methodologies of object-oriented inheritance [32, 2].

An overview of component-based client/server frameworks is provided in [19]. Sets of component-based distributed embedded systems facilitate coordinated interactions [5]. Redeployment of components across a network infrastructure results in improvements regarding service availability [20]. Alternatively, mobile devices can exchange software components in peer-to-peer networks in order to address user requests [25]. The exchange of software components in heterogenous hardware infrastructures might require adaptation of the control over the respective components or of their data. In [11], such an adaptation strategy is presented for transferring components among devices of varying degrees of computational power, e.g., from a desktop/server to a set of mobile devices. In particular, adaptation is realized by specific drivers that serve as middleware to translate the broadcast software components.

14.3 The LINDSAY Virtual Human

We started developing *LINDSAY Virtual Human* in May 2009 as a collaboration between the Department of Computer Science in the Faculty of Science and the Undergraduate Medical Education program in the Faculty of Medicine. The *LINDSAY* project currently has a demonstration room that replicates a medium size class room (20-30 students) with a large rear-projection screen. With *LINDSAY* we create, use and test (1) stereoscopic display technologies, (2) software for 3D content sharing across the internet, (3) motion capture technologies to develop seamless gesture-based user interfaces, and (4) wireless remote controls and touch interfaces based on iPhones, iPod touch devices and iPads. Figure 14.1 provides a snapshot of the different components of the *LINDSAY* project that its development team is currently working on.

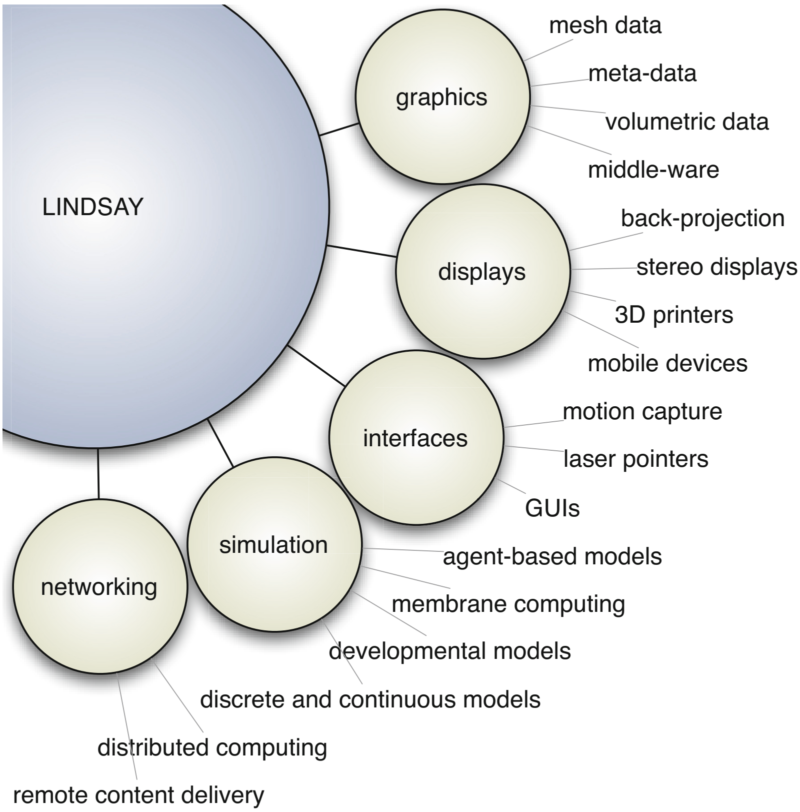


Fig. 14.1. As a truly interdisciplinary project, the *LINDSAY* system requires the integration of ideas, techniques and solutions from a wide range of fields within the disciplines of information and communication technologies.

14.4 LINDSAY Presenter

A typical scenario for an anatomy lecture for 1^{st} year students works usually like this: The anatomy instructor is standing in front of a class of a few hundred students, elaborating on the human skeleton, muscle, and nervous systems. The instructor is using a plastic model to explain the various muscles of the hand, draws with a pen on rubber gloves to illustrate tissue connectivity, or uses projected images on a screen in front of the class in the form of a slide presentation. This is not an ideal situation for the following reasons:

- Only very few students can actually follow the instructor's explanation, as they sit close enough to see which anatomical structures are being pointed out. Most of the students in the classroom are relegated to what

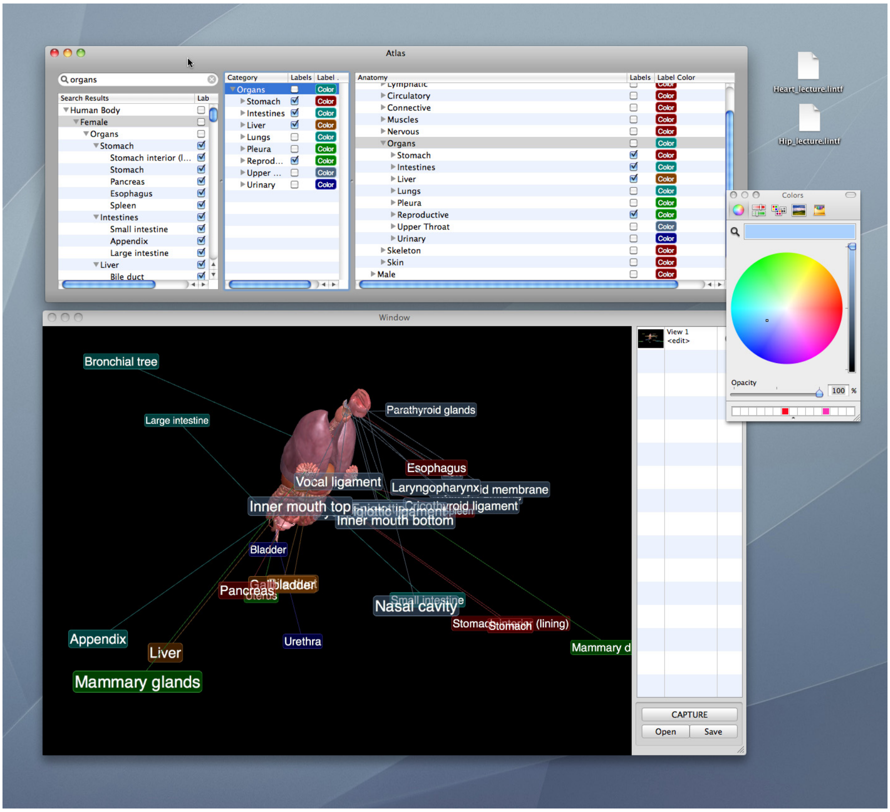


Fig. 14.2. The *LINDSAY* Presenter interface with its two main windows: (a) The top window displays the hierarchical list of the male and female anatomy atlas. (b) Below is the main display window, in which the currently selected anatomical structures are visualized.

they see on the projected slides, which they might (in many situations they don't!) have also on their laptop screens in front of them.

- The location of anatomical structures within the human body is best understood by getting a true feeling of their arrangements in 3D space. That is, it helps to be able to rotate the body, dissect to reveal hidden structures, and reassemble to get a true sense of each structure's exact position.
- The interrelationships between anatomical structures—such as organs, connective tissues, muscles, cardiovascular or lymphatic networks—is best revealed by, again, studying their points of connection and their distribution across the body.

Obviously, these aspects are hard to illustrate to students without being able to convey the true dimensionality, spatial arrangement and connectivity of anatomical structures. Furthermore, complex structures of the human body and their interrelationships reveal their functions much more easily by seeing them in action and in coordination with other components. Our *LINDSAY Virtual Human* tries to address exactly those issues.

14.4.1 Anatomy Atlas

The backbone of *LINDSAY* is an anatomical database, derived from a male and female anatomy model acquired from Zygote, Media Group, Inc. [35]. The model consists of a set of 3D surface files, organized in folders according to anatomical categories (vascular system, circulatory system, muscles, etc).

However, each file has labels which are often abbreviated and sometimes have no reference to anatomical categories of parts. Using anatomical atlases, we properly labelled and further categorized the parts, storing the data in an XML file. We converted the model to the COLLADA format, which we use as our data interchange format [10]. This facilitates the storage and transfer of not only static model data, but also rigging and information relevant to animation. We import and store the model data in our own binary format, along with the XML file containing the proper labelling and hierarchical organization of anatomical parts.

A snapshot of the user interface to the database is depicted in Figure 14.2. The atlas contains female and male anatomy, which is accessible via a hierarchical list. The selections through this interface determine which anatomical structures are displayed in the visualization window. Both single and multiple selections are possible, so that, for example, the cardiovascular system (heart and blood vessels) can be displayed in combination with the lymphatic network of the immune system.

The database is also searchable. Search terms can be entered in the field on the top of the left column of the atlas interface. Below this search field, the entries that match the query are displayed. From this result list, further selections can be made to add to the components displayed in the visualization window. The centre column serves as a common-item placeholder. Items from the atlas or search can be dragged in this middle table in order to keep only those components that are needed in a neat and arranged list.

Each entry in the atlas can be annotated with a label in the visualization window. This is illustrated in Figure 14.2, where many of the displayed inner organs are identified by their labels. The labels are arranged in the 3D space around the virtual human. Whenever the model is rotated, the positioning of the labels is adjusted to help with their readablity. As the labels are printed on a transparent background, it is usually easy to read most of them at the same time; if necessary, a small rotation of the anatomical structure would quickly reveal any hidden labels. This greatly improves legibility of the labels and gaining an intuitive sense of which label is associated with which anatomical parts. Organizing the labels and identifying the corresponding parts is also facilitated by different colours, which can be set through the colour chooser window.

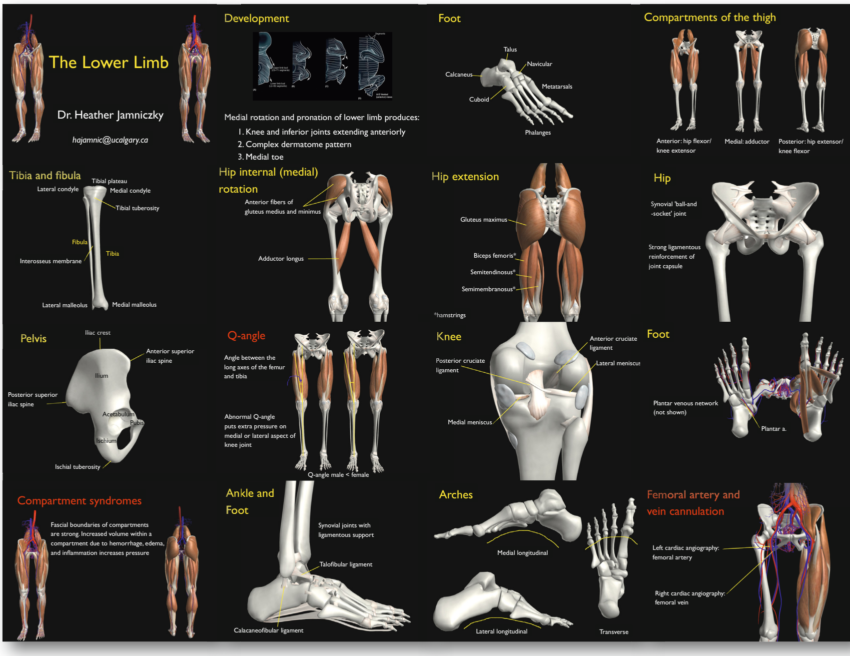


Fig. 14.3. Example slide set created with *LINDSAY Presenter*.

14.4.2 Interactivity

The 3D anatomy model can be rotated, moved, as well as scaled—so that structures from the whole body level to the organismal, cellular, and subcellular level can be inspected. These changes of perspective can be controlled using keyboard commands in combination with a normal 2D mouse and its control buttons. In addition, we have created a remote control application which runs on iPod touches, iPhones, and on a Pen-Smart pad.

We have also designed prototypical implementations of a gesture-based control interface, where gestures are identified on a video stream from a camera and then translated into navigational commands within the virtual human. Although they were generally well received by the anatomy instructors, they could not (easily) allow for a high degree of interaction, comprising numerous commands. We have conducted experiments with laser pointers that we can use as drawing devices on the projection screen. Here, again, a video camera behind the screen is identifying the position of the laser beam on the screen, translating these into the proper coordinates within the main display window. In the meantime, we have replaced the laser pointer by an iPod touch and iPad, where the drawing is now performed on the touch surface. Figure 14.4(a) shows the navigation screen of our remote control application on an iPhone. Moving a single finger across the touch interface relocates the virtual human in its x-y-plane. Moving two fingers apart or close to each other will increase or decrease the zoom level, respectively. Rotating the iPhone/iPod around its central and lateral axis will make the virtual human turn accordingly. A vertical two-finger swipe moves a customizable cutting surface in and out of the virtual body, which allows one, for example, to see through the ribs, skin and muscles into the inside of the heart. This combination of touch gestures and rotation makes the iPhone/iPod an intuitive remote control device for instructors, who should not need to use the keyboard or mouse for navigation of and around the virtual human.

The iPhone/iPod application can also be switched to atlas mode, where a replica of the hierarchical anatomy ontology is available through the touch interface. Figure 14.4(b) illustrates how this touch interface works. As the iPhone/iPod screen is rather small, it is not practical to display the hierarchical anatomy list all at once. Rather, we opted to show only a single hierarchical level per screen. Selections of anatomical structures are made by double-clicking. Any selections made through the iPhone/iPod interface are immediately replicated on the atlas interface in the main application (Fig. 14.2).

Both through the main *LINDSAY Presenter* and its wireless touch devices, the user can draw simple shapes and lines to enhance or highlight the 3D model currently on display. Custom labels can also be added, which can optionally be connected to specific anatomical structures. These labels are in addition to the already built-in annotations, which display the names of the anatomical structures in the atlas.

14.4.3 Creating 3D Slides

As *LINDSAY Presenter* is meant to be used as a presentation tool of human anatomy, we have included an easy mechanism to build sequences of 3D slides. This is similar to the way standard slide presentation software, such as Powerpoint© or Keynote© work; after having created a set of slides, one can progress from slide to slide to enhance an oral presentation. In

(a) (b) (c)

Fig. 14.4. The *LINDSAY Presenter* remote control applications running on iPhone and iPod touch devices: (a) The navigation interface allows the virtual body to be moved, scaled, and rotated. Different styles of cuts can 'open' the body. (b) The interface for the atlas gives access to all anatomical structures. (c) Three categories of anatomical parts are selected (indicated by the pink background), which are instantly displayed in the main window.

LINDSAY Presenter slides are actually scene descriptions of the 3-dimensional contents; therefore, we refer to these scenes as "3D slides". At any point a snapshot of the current 3D scene can be taken by selecting the `CAPTURE` button at the bottom right corner of the main display window (Fig. 14.5). This then adds a thumbnail in the 3D slide column to the right of the display window. Flipping between 3D slides creates a smooth, animated transition between the associated scenes. A set of such 3D slides can be saved (`Save`) and later loaded (`Open`) into the *LINDSAY Presenter* for subsequent presentation, where the 3D slides can be used as guidance. Figure 14.6 gives examples of such a sequence of 3D slides, here illustrating the anatomy of the female hip. It is worth mentioning, that each 3D slide is, of course, completely interactive, as the "slides" are not simple screen captures of the current 3D contents, but contain the actual scene graph of the current display.

14.4.4 Volumetric Data Integration

We acquired image slice data (Fig. 14.7(a)) of the human body from the Visible Human Project [22]. Using the OpenGL Shading Language, we can render a cut plane of the head from any orientation (Fig. 14.7(b)). Combining the shape of the head with a cut plane, an arbitrary cross-sectional view from any position or orientation can be achieved as demonstrated on the iPhone (Fig. 14.7(c)). We also render using multiple layers to visualize all

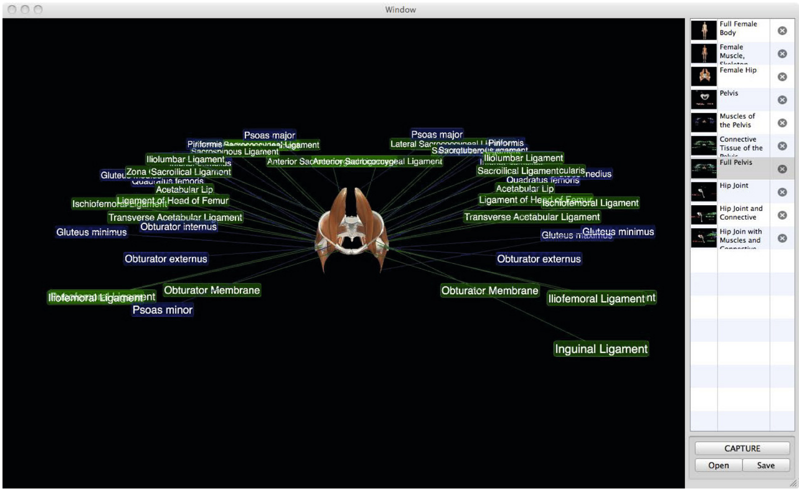


Fig. 14.5. 3D Slides in *LINDSAY Presenter*: Demonstration of a typical setup for a lecture, here, as an example, on muscle anatomy. See Figure 14.6 for close-ups of selected 3D slides.

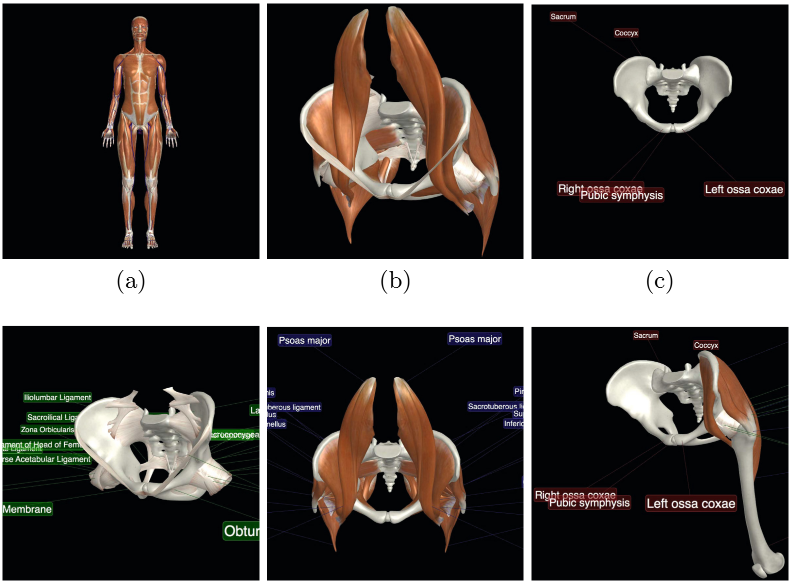
the slices simultaneously to achieve a volumetric visualization; for this we use the shader again to allow for an arbitrary viewing orientation(14.7(d)). With the application of transparency and colour filters, inner structures can be revealed or isolated (Fig. 14.7(e), (f)).

The Zygote model [35] is stylized and representational, designed to be illustrative of the major structures and well suited for educational purposes. For example, arteries are coloured solid red and veins are coloured solid blue. The Visible Human slice data on the other hand is from photographs taken of actual slices of a cadaver, and often individual organs aren't as readily discernible and surface boundaries not always obvious.

Rendering the data sets together allows for further exploratory abilities. Figures 14.7(d-h) show the slice data superimposed over the 3D model data. Thus, direct comparisons can be made between the representational visualization and the realistic and physically accurate volumetric datasets. In Figures 14.7(g) and 14.7(h) the cut plane reveals the inner structures of the 3D model, while the slice data remains simultaneously overlaid.

14.5 LINDSAY Composer

One of the key objectives for *LINDSAY* is to support the creation of computational models in real-time and combine this with 3-dimensional, highly



(a) (b) (c)

(d) (e) (f)

Fig. 14.6. Examples of 3D Slides in *LINDSAY Presenter*: A typical sequence of 3D scenes that would mark keypoints of navigation through the *LINDSAY* anatomy. Smooth, animated transitions are generated to blend between slides: (a) Female muscle skeleton, (b) female hip, (c) pelvis, (d) connective tissue of pelvis, (e) pelvis muscles, (f) hip joint with muscles.

interactive visualizations of physiological processes related to the human body. *LINDSAY Composer* is a primarily graphics-based programming environment, through which complex biological processes can be compiled using simple drag-and-drop interfaces. Available "building blocks" for such compositions are, for example, fluid fields, cameras, anatomical objects, protein shapes, or cell types. A time line helps to trigger and keep track of events within the simulation, which then allows one to run and replay simulations as needed (compare Fig. 14.12).

Similar to first-person 3D computer games, the *LINDSAY Composer* creates an environment that immerses the user into a virtual 3-dimensional world, with the context of anatomical structures and processes. Different from prerecorded contents, the underlying computational engines, in combination with the dynamic scene composition, allow for interactive explorations that are guided by the user's intuition and curiosity. Software engineering methodologies, hardware advancements and agent-based models turn simulations into a highly interactive multimedia experience.

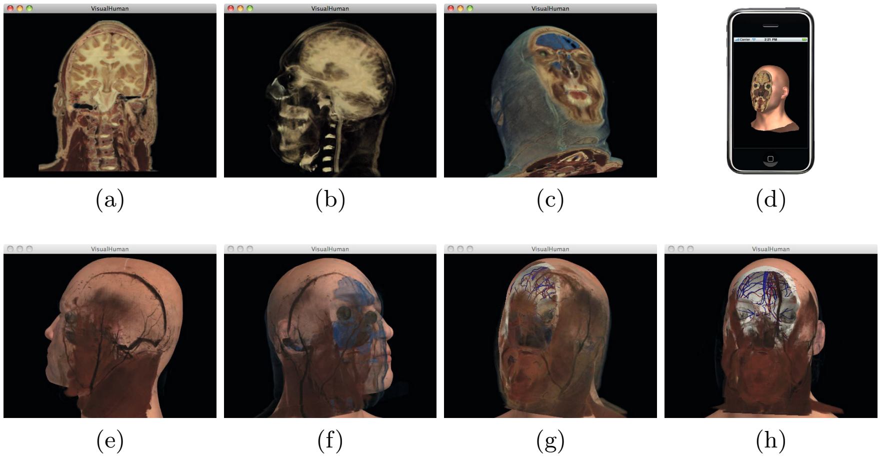


Fig. 14.7. Integration of volumetric data from the Visible Human with surface model data. (a-c) Renderings of the volumetric data from the Visible Human data set within *LINDSAY Presenter*; (e-h) the anatomical model data is superimposed on the volumetric data set. (d) The renderings work on both desktop and mobile devices.

14.5.1 The Computational Framework

LINDSAY is being built as a fully functional, 3-dimensional model of male and female anatomy and physiology. Our objective is to use the virtual human to illustrate, for example, which muscles are involved when we run or when we hold a pen. Similarly, we can illustrate and investigate the functional structures and defense processes involved in the human immune system. The immune system, in particular, encompasses complex processes across a wide range of scales: from the organismal level (e.g., the thymus) to the networked lymphatic system (including lymph nodes) down to the cellular (B-, T-cells) and the sub-cellular levels (gene regulation in response to pathogens).

Consequently, *LINDSAY*'s modeling framework aims to integrate mathematical and computational models across scales. For example, we use fluid dynamics to simulate blood flow at the whole body level. Using interactive zooming, we can get inside a blood vessel, dive into a capillary or hop onto a blood cell to carry us through the blood stream.

At this level we switch to agent-based simulations, where "bio-agents" (i.e., cells, proteins and other molecular structures) are being tracked within the simulated 3D virtual body space. Physics engines, mainly used for computer games, have been adjusted to provide realistic and yet quickly computable scenarios that replicate physiological processes within the human body.

As part of the LINDSAY Virtual Human project, we have developed a component-based computational framework that allows the utilization of various formal representations, computation engines and visualization technologies within a single simulation context. For our agent-based simulations, the graphics, physics and behaviours of our interacting entities are implemented through a set of component engines. We have developed a light-weight client/server component, which spreads its siblings in the system's component hierarchy over a wireless or wired network infrastructure.

14.5.2 Agent-based Modelling

Agent-based models play an increasingly dominant role for the modeling of biological and social systems [17, 33, 24]. We see the most promising potential in agent models that incorporate swarm intelligence techniques [7, 9], as these result in more accurate and realistic models. The use of agents as independent and interacting entities is particularly crucial when spatial aspects play a key role in defining patterns of interaction, in understanding their emergent properties, and in helping to shed light on the inner workings of physiological processes.

All biological systems, such as the human body, are inherently driven by interaction processes in a 3-dimensional world. Therefore, in order to capture physiological processes within our *LINDSAY Virtual Human*, we utilize swarm-based, 3-D simulations which exhibit a high degree of self-organization, triggered by relatively simple interactions of a large number of 'bio-agents' of different types.

The *LINDSAY* human body is a perfect example that allows for middle-out modeling [31]. Other models, which we have worked on before and are currently incorporating into *LINDSAY*, include the study of chemotaxis within a colony of evolving bacteria [13, 23], the simulation of transcription, translation, and specific gene regulatory processes [16], as well as studies of affinity and cooperation among gene regulatory agents for the λ switch in *E. coli* [15].

14.5.3 Component Architecture

LINDSAY Composer is written in Objective-C, but is also compatible with C, C++, Python and any other scripting languages that can interface with the Objective-C runtime system. The core framework of *LINDSAY Composer* is as minimal as possible. Having such a simple framework makes it very easy for developers to extend the system with plugins that add new component types, new component engines, and even new user interfaces. Without great

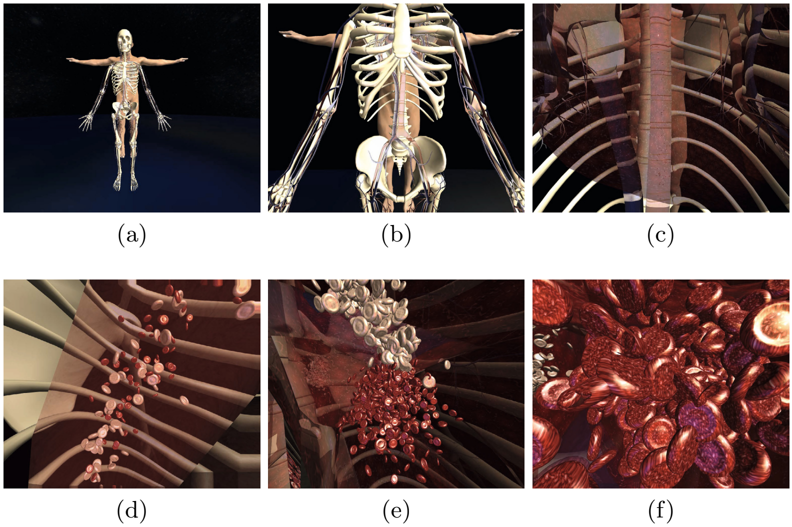


Fig. 14.8. Computational modeling of physiological processes across scales by example of the circulatory system: (a) full body view, (b) close-up with upper skeleton, (c) inside the rib cage, (d) approaching the main artery, (e) inside the main artery with red and white lymphocytes, (f) close-up of a cluster of lymphocytes.

efforts, components can collaborate with each other due to the simplicity of a component's API.

In *LINDSAY Composer* a simulation is represented as a hierarchy of components that encapsulate state and behaviour. A component may be registered with a computational engine (Fig. 14.9). For instance, one might have a physics engine managing rigid body interactions, those rigid bodies would be represented in the simulation by physics components. It is not necessary for a component to be registered with a component engine, as is the case with a component that holds only position and orientation data, which we call a transform component.

Components can query their parents for siblings of a given type. Components can also query the entire simulation for components that match a given expression. A common use for this mechanism is for sibling components to share a transform component. For example, one might have a physics component updating the transform, while a rendering component draws a 3D mesh relying on the very same transformation information. In another instance, a camera may be manipulated by a user interaction component that updates a transform component's orientation and position. A graphics component would then adjust the view in the direction indicated.

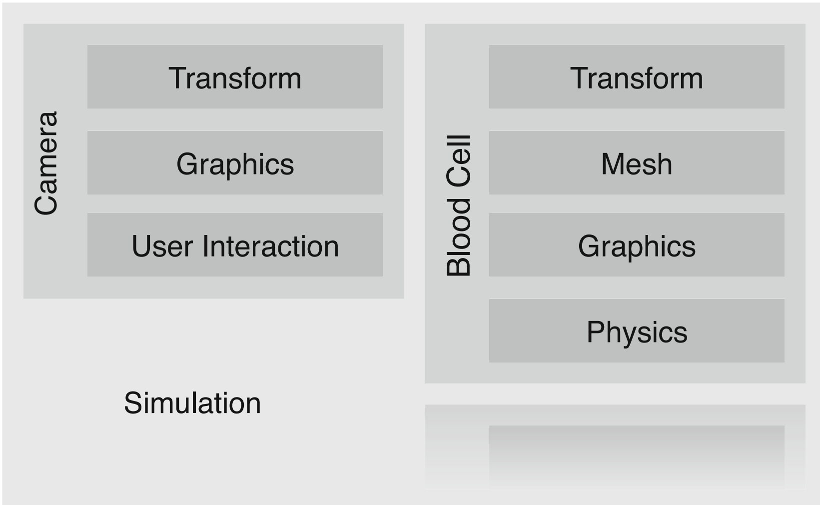


Fig. 14.9. A component hierarchy typical of most simulations. Boxes represent components, the embedding of boxes within each other is representative of the component hierarchy, shading is used to exemplify this.

LINDSAY Composer uses a number of engines to form the core computational framework (Fig. 14.10). For instance, a *physics* engine manages rigid body interactions and dynamics, a *graphics* engine renders the simulation to the screen, a *networking* engine handles the distribution of components to other nodes in a network, and an *interaction* engine allows the user to interact with the simulation via mouse and keyboard or multi-touch enabled devices. Each simulation engine iterates through the components it manages to update them. Figure 14.10 shows how components are associated with different component engines while still being part of a component hierarchy.

Especially important for the interactive simulation control as well as for collaborative classroom experiences are the networking components of the *LINDSAY Composer* [30]. A server component is anchored into the simulation's component hierarchy which replicates its siblings and sends them over a network to one or more client components. There can be multiple servers and clients within a single simulation. This allows for distribution of a shared simulation to a number of client devices, which is a typical scenario in a classroom setting. It also allows for interaction with a simulation via mobile devices such as the iPod Touch, iPhone or iPad.

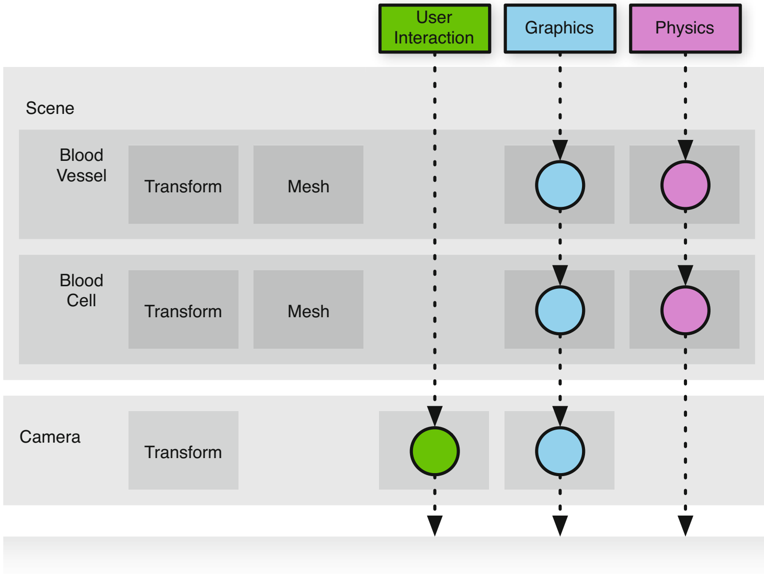
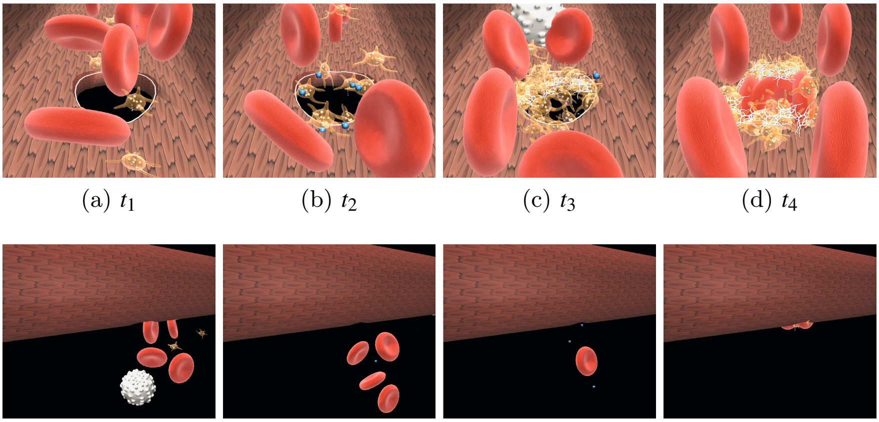


Fig. 14.10. Example architecture of component engines: The square boxes at the top are simulation engines for user interaction, graphics, and physics. Elements in the simulation, such as blood vessels and blood cells, contain components for their rendering, interaction, and physical properties. Those components register with the respective engines (denoted by the arrows). Other components, such as transform and mesh, are not registered with a simulation engine. As in Figure 14.9, the nesting of boxes (and circles) corresponds to a components place within the component hierarchy.

14.5.4 Graphical Programming Interface

In this section we explore the *LINDSAY Composer* user interface. Figure 14.11 shows a blood clotting simulation that we built with *LINDSAY Composer*. The simulation displays two separate views of the same model at different time steps [34]. At the top one can see red and white blood cells, fibrinogens, fibrins, platelets, and other messenger molecules. All of these are implemented as separate agents, that interact with one another and with the blood vessel wall, which is lined by endothelial cells. Over time, one can see that the wound site is eventually filled by platelets and fibrins, which stops the simulated bleeding. The bottom row shows the same simulation from a perspective outside of the blood vessel, which is embedded in the circulatory system of the virtual body (compare Fig. 14.8).



(a) t_1 (b) t_2 (c) t_3 (d) t_4

(e) t_1 (f) t_2 (g) t_3 (h) t_4

Fig. 14.11. The blood coagulation simulation at different time steps ($t_1 < t_2 < t_3 < t_4$). The process is observed from two different perspectives: inside (a-d) and outside (e-h) of the vessel. Different views are defined by cameras that can be navigated through and are located within the simulations.

Building a simulation in *LINDSAY Composer* is achieved via a drag and drop interface. Users can drag components from a library of component types into the simulation view, where they will appear immediately if they have a graphical representation. Figure 14.12 shows a typical screen display during the modelling process of a simulation.

In the toolbar, at the top of the window, we see controls for the camera, for bringing up the scene inspector (Fig. 14.13), for the frame rate, and for hiding the template library and timeline views. The user may directly interact with the scene by way of a 3D navigation and selection interface. Navigation and selection is performed via mouse and keyboard, or via networking components through mobile devices with touch interfaces.

In Figure 14.12, the component library is displayed on the left-hand side. From here, components are dragged into the simulation view on the right. Directly below the simulation view is the timeline view. We use a timeline to control the lifetime and properties of components within a simulation. This allows for the dynamic control of a simulation. The horizontal bars in the timeline correlate with the objects that exist in the simulation.

Overlaying the topmost time-bar we see an interpolation graph, which is represented as a component within the simulation. In our blood clotting simulation this graph corresponds to the velocity of the blood within the blood vessel, hence we can simulate the pulsing action of the heart on blood within a blood vessel. These interpolation components can be applied to any compatible property of any component, or set of components within the simulation.

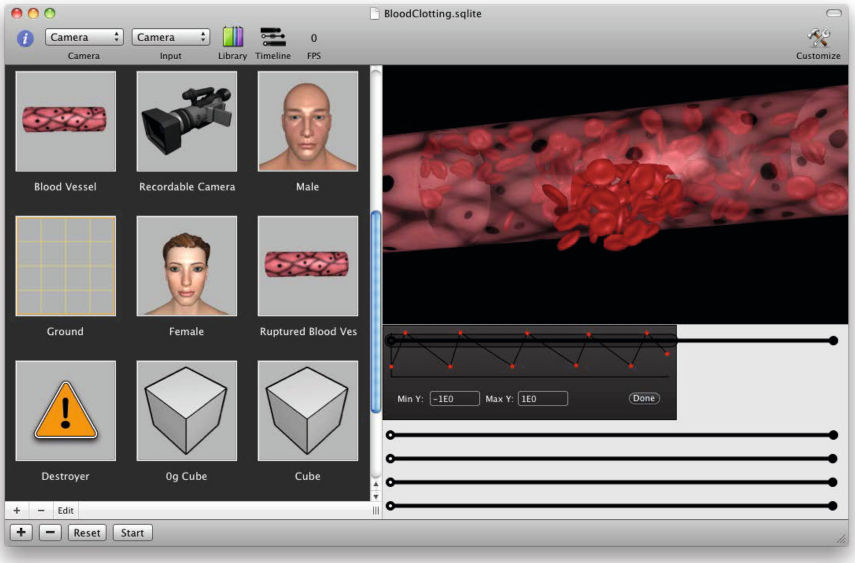


Fig. 14.12. Components are dragged from the left into the rendering view on the right to compose a simulation. An overview of the interaction processes is provided in a configurable timeline (bottom-right). The centred overlay window shows an interpolation component that directs the blood flow.

Another important part of the *LINDSAY Composer* interface is the inspection of templates and of the hierarchy of a simulation (Fig. 14.13). We dynamically generate the interaction dialogs for a component at runtime. This user interface allows for the realtime inspection and editing of properties as the simulation runs. We also allow for preprogrammed interfaces by checking for their existence at runtime.

14.6 The Educational Perspective

Medicine is replete with complex information spaces. Integrating physiology and pathology, as well as functional anatomy, within true anatomical displays—as opposed to full cartoon images of the body—gives users the opportunity to learn multiple content pieces at the same time. From an educational perspective, we are approaching an application of cognitive flexibility theory and constructivism within a simulation environment that most closely represents true (not cartoon) anatomy.

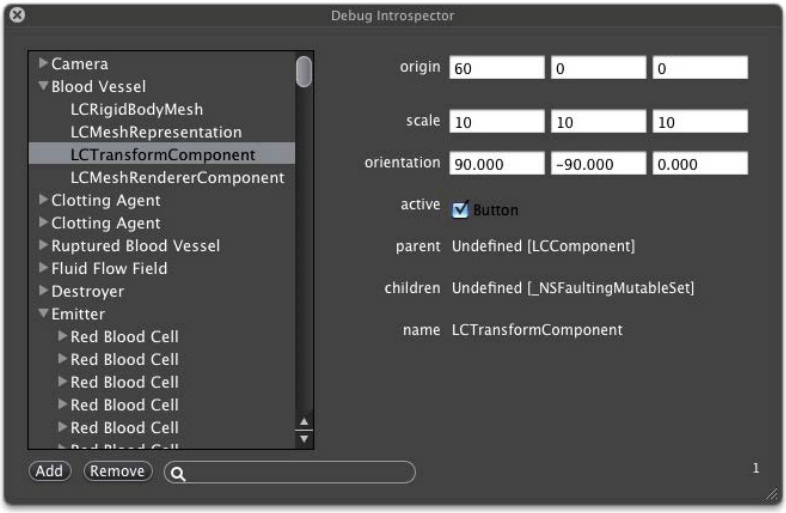


Fig. 14.13. (a) The *LINDSAY Composer* GUI for inspecting the scene, its hierarchy, and its components.

Putting the tools in the hands of the users allows them to build their knowledge and to construct pathways for their own learning. Each learner is able to pace their learning, and/or create a physiological or anatomical environment that suits their personal learning needs. Faculty is able to construct, play, pause, rewind, re-create and deconstruct the learning environment in order to emphasize, assess, or modify learning tempo and goals to meet learners' needs.

14.7 Current and Future Work

Interactive Classroom

After anatomical and/or physiological scenarios have been created using *LINDSAY Composer*, one can use *LINDSAY Presenter*to run, demonstrate and explore these compositions. We are currently working with instructors from medicine, nursing, kinesiology, and veterinary medicine to turn our software into an effective presentation toolset for the classroom. Using the described client/server architecture, we are now able to create multiple visualization and remote control components that are shared on a wireless network. This lays the foundations to create innovative scenarios for inquiry-based learning and teaching. Using iPods and iPads, we deliver visualizations to wireless devices, which, in turn, can be used to inspect and control different aspects of a shared physiology model, which runs on a powerful simulation server.

Remote Learning and Content Sharing

Building an infrastructure for immersive visualization revolutionizes the way human anatomy and physiology is being taught. Such 3D contents can be shared worldwide across the internet by using 3D content sharing technologies, where multiple users can see, "grasp" and interact with 3-dimensional anatomical objects—from organs to cells—and observe the unfolding of simulated physiological processes. This means, contents produced in one location can be distributed to other educational and research institutions across the globe.

14.8 Conclusions

We described the *LINDSAY Virtual Human* project, gave details on its programming frameworks, as well as its display, visualization, and interaction technologies. We also shared our initial experiences about how to bring computer modeling, 3D visualization and human-computer interaction technologies into medical classrooms. More information on the *LINDSAY Virtual Human* project is available on our website:

`http://lindsayvirtualhuman.org`.

Acknowledgments. We would like to thank the Department of Undergraduate Medical Education, Faculty of Medicine, at the University of Calgary for supporting the *LINDSAY Virtual Human* project. We also greatly acknowledge financial support for the *LINDSAY* project by the Lindsay Leigh Kimmett Memorial Foundation (lindsaykimmett.org). Part of this project is supported by the Natural Sciences and Engineering Research Council of Canada. The anatomy model data sets are provided by Zygote Media Group, Inc. (www.zygote.com).

References

1. Entity systems are the future of MMOG development (September 2007), `http://t-machine.org/index.php/2007/09/03/entity-systems-are-the-future-of-mmog-development-part-1`
2. Evolve your hierarchy (January 2007), `http://cowboyprogramming.com/2007/01/05/evolve-your-heirachy`
3. Harvard BioVisions (January 2012), `http://multimedia.mcb.harvard.edu`
4. Ackerman, M.J., Yoo, T., Jenkins, D.: From data to knowledge - the visible human project® continues. In: MEDINFO 2001, pp. 887–890 (2001)

5. Årzén, K., Bicchi, A., Dini, G., Hailes, S., Johanesson, K., Lygeros, J., Tzes, A.: A component-based approach to the design of networked control systems. European Journal of Control 13(2), 261–279 (2007)
6. Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riß, S., Sandor, C., Wagner, M.: Design of a component-based augmented reality framework. In: International Symposium on Augmented Reality, Los Alamitos, CA, USA, p. 45 (2001)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, New York (1999)
8. Burleigh, I., Suen, G., Jacob, C.: Dna in action! a 3d swarm-based model of a gene regulatory system. In: First Australian Conference on Artificial Life, ACAL 2003, Canberra, Australia (2003)
9. Camazine, S., Deneubourg, J.-L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organization in Biological Systems. Princeton Studies in Complexity. Princeton University Press, Princeton (2003)
10. COLLADA.org. Collada - digital asset and fx exchange schema (April 2010), `http://www.collada.org`
11. de Lara, E., Wallach, D.S., Zwaenepoel, W.: Puppeteer: Component-based adaptation for mobile computing. In: USENIX Symposium on Internet Technologies and Systems (2000)
12. Heineman, G.T., Councill, W.T.: Component-Based Software Engineering. ACM Press (2001)
13. Hoar, R., Penner, J., Jacob, C.: Transcription and evolution of a virtual bacteria culture. In: IEEE Congress on Evolutionary Computation. IEEE Press, Canberra (2003)
14. Hunter, P.J., Borg, T.K.: Integration from proteins to organs: the physiome project. Nature Reviews: Molecular Cell Biology 4, 237–243 (2003)
15. Jacob, C., Barbasiewicz, A., Tsui, G.: Swarms and genes: Exploring λ-switch gene regulation through swarm intelligence. In: IEEE Congress on Evolutionary Computation, CEC 2006 (2006)
16. Jacob, C., Burleigh, I.: Biomolecular swarms: An agent-based model of the lactose operon. Natural Computing 3(4), 361–376 (2004)
17. Johnson, S.: Emergence: The Connected Lives of Ants, Brains, Cities, and Software. Scribner, New York (2001)
18. Leavens, G.T., Sitaraman, M. (eds.): Foundations of component-based systems. Cambridge University Press, New York (2000)
19. Lewandowski, S.M.: Frameworks for component-based client server computing. ACM Comput. Surv. 30(1), 3–27 (1998)
20. Mikic-Rakic, M., Malek, S., Medvidovic, N.: Improving availability in large, distributed component-based systems via redeployment. Component Deployment, 83–98 (2005)
21. Mitsuhashi, N., Fujieda, K., Tamura, T., Kawamoto, S., Takagi, T., Okubo, K.: BodyParts 3D: 3D structure database for anatomical concepts. Nucl. Acids Res. (2008), gkn613
22. N. L. of Medicine. National library of medicine: The visible human project (April 2010), `http://www.nlm.nih.gov/research/visible/visible_human.html`
23. Penner, J., Hoar, R., Jacob, C.: Bacterial chemotaxis in silico. In: First Australian Conference on Artificial Life, ACAL 2003, Canberra, Australia (2003)

24. Pogson, M., Smallwood, R., Qwarnstrom, E., Holcomb, M.: Formal agent-based modelling of intracellular chemical interactions. BioSystems (2006)
25. Roussain, H., Guidec, F.: Cooperative component-based software deployment in wireless ad hoc networks. Component Deployment, 1–16 (2005)
26. Seifert, M.F.: Visible human projects special issue. Clinical Anatomy 19(3), 191–289 (2010)
27. Sensen, C.W., Soh, J.: CAVEman, An Object-Oriented Model of the Human Body. In: Sensen, C.W., Soh, J. (eds.) Advanced Imaging in Biology and Medicine, vol. II, pp. 289–300. Springer, Heidelberg (2009)
28. Stoy, C.: Game Object Component System. In: Game Programming Gems 6, pp. 393–403. Charles River Media (2006)
29. Tietze, D. A.: A Framework for Developing Component-based Co-operative Applications. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany (2000)
30. von Mammen, S., Davison, T., Baghi, H., Jacob, C.: Component-based networking for simulations in medical education. In: MediaWiN 2010: IEEE Workshop on Multimedia Applications Over Wireless Networks, Riccione, Italy. IEEE Computer Society, IEEE Press, Washington (2010)
31. Walker, D.C., Southgate, J.: The virtual cell–a candidate co-ordinator for 'middle-out' modelling of biological systems. Briefings in Bioinformatics 10(4), 450–461 (2009)
32. Wilson, K.: Game object structure: Inheritance vs aggregation (July 2002)
33. Wolfram, S.: A new kind of science. Wolfram Media Inc., Champaign (2002)
34. Yazdanbod, I., Marcus, S.: An agent-based simulation of blood coagulation processes. The Journal of Undergraduate Research in Alberta 1(1), 13–18 (2010)
35. Zygote Inc. Zygote human anatomy 3d model (April 2010), `http://www.zygote.com`