The CoSMoS Process, Version 0.1: A Process for the Modelling and Simulation of Complex Systems

University of York Technical Report YCS-2010-453

Paul S. Andrews   Fiona A. C. Polack   Adam T. Sampson
Susan Stepney   Jon Timmis

March 2010

Contents

Chapter 1

Introduction

The CoSMoS project[1] aims to build capacity in generic modelling tools and simulation techniques for complex systems, to support the modelling and analysis of complex systems, and to help design and validate complex systems.

This report summarises the achievements of the project to date (March 2010) in developing a generic process for complex system simulation, providing guidelines and techniques that enable the construction and exploration of simulations for the purpose of scientific research. The CoSMoS process is necessarily an interdisciplinary endeavour between scientists who study a particular domain, and engineers/developers who construct simulations to enable the study of that domain. Together, both the scientists and engineers are involved in open-ended scientific research; the simulations are used as a tool to elaborate and explore science in a wider context.

It is inevitable in a report such as this that the description appears prescriptive. However, the overriding CoSMoS philosophy is flexibility. Subsequent reports will consider how the process has been applied to the project simulation exercises, and will better demonstrate the adaptability of the process. A brief summary of the current scientific case studies can be found in [10]. These examples use the evolving process, contribute to the evolution, and demonstrate how the general process can be tailored.

Chapter 1 of this report presents aspects of the background and motivation for CoSMoS, drawing on various published papers. Chapter 2 outlines the CoSMoS process, in terms of phases, products, activities and roles. This defines the terminology of CoSMoS development, identifying generic and phase-specific aspects. Chapter 3 considers how the activities might relate to the phases of the CoSMoS process, highlighting the core aspects of the CoSMoS process and the things that are optional.

[1]EPSRC grants EP/E053505/1 and EP/E049419/1, www.cosmos-research.org, 2007-2011

1.1 The What and Why of the CoSMoS Process

The CoSMoS project is looking at two uses of simulation. The first is in support of scientific research. The second is the engineering use of simulation, specifically in the development of swarm robotics. The process of developing these simulations differs in detail and emphasis, though the underlying guidelines and philosophy are the same. Most of this report is written in relation to the scientific simulations.

A process is a way of doing something: it is a structured set of tasks to achieve one or more goals. The CoSMoS process will consist of a set of generic structuring concepts, a pattern library for applying these concepts, the philosophy behind working with complex systems simulation, and examples of the application of the process. It will incorporate all the advice and structure we provide in this document to aid the construction of complex system simulations. The process for engineering scientific simulations is being extracted from a series of case-studies, developed with scientists from a broad range of disciplines (immunology, ecology, and, more recently, sociology), see [10]. Here we focus on introducing a rationale for our approach plus the generic structuring elements of the CoSMoS process.

We emphasise that the CoSMoS process is a flexible approach [10], adaptable both to a variety of simulation problems and to changing circumstances during simulator construction and use. The application of the CoSMoS process in any context should be tailored to suit the criticality and intended impact of the underlying research (or engineering) project.

It has been noted by authors such as Humphreys [5] that many areas of contemporary science are now driven by computational methods, and as a consequence the development of these methods has, or needs to, assume a new direction. A common computational method, with widespread use across numerous disciplines is, *computer simulation*. To our knowledge, little work exists that provide guidelines on how best to develop and use simulations of complex scientific systems. It is this hole that the CoSMoS process attempts to fill.

Computer simulation is a potentially valuable tool in many areas of scientific enquiry. In relation to scientific exploration, CoSMoS is addressing the engineering of simulations to support theory exploration, hypothesis generation, and design of real-world experimentation. Our approach focuses on agent-based simulations in which the agents of a system are directly modelled as computational processes, allowing complex behaviours to emerge naturally from interactions in a simulated environment.

However, we believe that our findings are applicable to other simulation approaches; see [13] for more details.

We study complex systems both as a means to understand the world around us and to apply this understanding to solve problems. When we study a system, we naturally model our understanding of that system. In many cases this model is informal – a mental model that may get represented as a block of text. The model makes many implicit assumptions that may be difficult to fully justify[2].

In terms of exploring complex systems through simulation (the purpose of the CoSMoS process), a traditional scientific approach is augmented by engineering an artefact (the simulation) that can contribute to the understanding of the system of study. In building simulations we add an extra layer of complexity, in trying to represent our system understanding in an explicit and implementable form. To have confidence that the simulation can actually tell us something about the real domain we must be principled and make explicit where the engineered system has come from. The CoSMoS process aims to achieve a principled approach, involving explicit models and phases that support close interaction of domain scientists and simulation engineers. The simulation is thus exposed to review and challenge, and presents scientifically-reproducible work.

1.2 Complex Systems and CoSMoS

Simulation is "the technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus, esp. for the purpose of study or personnel training"[3]. Complex systems simulation can be realised in software (constructing a computer simulator) or in hardware (for example, using robots).

Complex system has many definitions. The CoSMoS process is aimed at systems that are complex in the sense of having elaborate behaviour at a high level that is the consequence of many simple behaviours at a lower level. The high-level behaviour cannot be deduced as a simple combination of low-level behaviours. Components of the complex system interact with and through an environment. The components and the environment are thus critical elements of the simulations (see [12, 11]).

[2]We consider an assumption as any kind of abstraction, simplification, axiom, idealisation or approximation made, whilst asking scientific questions or engaging in engineering.

[3]Oxford English Dictionary

The CoSMoS process can be used for:

1. Exploring, understanding, or describing complex (often non-intuitive) system behaviours;
2. Engineering complex behaviours in simulation.

The second use, engineering of complex behaviours, necessarily requires understanding of complex behaviours (the first use). A challenge for developers is how and when to obtain the necessary understanding. Ongoing work on the CoSMoS process will be addressing this in part, through capturing patterns of complexity and ways of developing the systems that display complexity. A body of work on understanding complex systems would thus be (one of) the domains of an instantiation of the CoSMoS process focusing on the engineering purpose, above.

The two uses of the CoSMoS process can be elaborated by considering scenarios in which each would be applicable:

1. Understanding complex systems behaviours:
   (a) Simulating real-world complex phenomena e.g. from biology, chemistry, physics, sociology.
   (b) Simulating complex systems "themes": looking at general properties or emergent characteristics of complex systems that transcend subject categorisation. For example, complex behaviours such as swarming, flocking, schooling; behaviours driven by environmental resource acquisition and exploitation.
2. Engineering complex behaviours:
   (a) Engineering a simulation to explore algorithms that rely on complexity. For example, the CoSMoS process could be applied to engineering a simulation of a bio-inspired algorithm such as an artificial immune system in the context of its application area. This would explore the potential behaviours of the application, and possibly provide bootstrapping for the artificial immune system (see [8, 9] for an approach influenced by, but independent of, CoSMoS).
   (b) Engineering a complex system, such as a swarm robotics system, that can manifest complex behaviours analogous to those observed in natural complex domains.

1.3 Models and Simulations

A *model* is an abstraction that is made to aid understanding or description of something. Polack et al [13] (like Jackson [6]) identify two orthogonal modelling goals with respect to complex systems: description and definition: "For a complex emergent system, a descriptive model might capture aspects of the observed high-level behaviour; in modelling natural systems, scientists use models to capture what they observe. A definitional model is more typical of conventional engineering – it expresses required characteristics of a system at an appropriate level of abstraction. A definitional model can be refined, translated and analysed, to improve understanding of system characteristics, and, in engineering, to support construction of an artificial system."[13]. The following is a summary of the review of modelling and simulation in [13], focusing on requirements for modelling complex systems. It is included here as it encompasses much of the motivation for CoSMoS.

To model a complex system, the model (or simulation) must capture structures within and among components; protocols for communication among components; and potential state changes. A simulation models system dynamics, but conventional model views are static. For example, diagrams may capture the state of a system or prescribe possible histories of a system. We also need to understand the layered processes that determine a particular complex system, and this is a motivation for simulation using agents: such models can express the characteristics of multiple instances of low-level systems, and may reveal emergent characteristics at the high level; they can represent the context of systems, in terms of space, time and relevant environmental features. Modelling requires clarity: models might express characteristics of the domain (the natural system that is being simulated) or characteristics of the engineering design, or aspects of the implementation, but each such (set of) models needs to be for one explicit purpose. There is a need to understand and express correspondence between the different sorts of model (see [13]).

The considerations noted so far are well known in engineering and have been used in complex system simulation work (see [15, 2, 16]). CoSMoS is also concerned with issues relating to dimensionality and scale, and patterns are being developed to address these.

Other motivations for the CoSMoS approach relate to more-generally desirable features of models. Of these, modifiability is essential, both in design and execution: "It is highly desirable that modifications in one view or instance of a model are reflected (automatically) in other views. In all areas, modification is used to adjust models to meet some external criteria (e.g. realism, customer requirements, etc). In engineering, modification also means *translation* from an abstract model to an implementation level, or between notations that are in some sense equivalent. The problems of consistency under modification have challenged designers for many years, and are exacerbated by inconsistent or non-integrated modelling tools and ill-defined notations."[13].

In the CoSMoS process, much of the engineering effort is undertaken to support modifiability, i.e. constructing a simulation platform that is modified according to current requirements. A working assumption in all the phases (chapter 2) is that a simulation platform is required, through which a series of simulations can be instantiated and run. This is essential in the scientific-research context, where CoSMoS provides a platform for *in silico* experimentation: we are explicitly not interested in one-off throw-away simulations. Similarly, in the engineering context (the second use of CoSMoS outlined in section 1.2), the issue of understanding as a prerequisite of engineering complex behaviours is likely to require iterated simulation.

1.4 Simulations as Scientific Instruments

Humphreys [5] identifies three ways in which scientific instruments enhance the range of our natural human abilities:

Extrapolation : the extension of an existing modality. For example, vision is extended via a microscope.

Conversion : the conversion of a feature from one mode to another. For example, a visual display on a sonar device. Conversion is often used in conjunction with extrapolation and augmentation.

Augmentation : accessing features of the world that we are not naturally equipped to detect in their original form. Examples include the detection of magnetism, elementary particle spin etc.

The three enhancements represent a spectrum as shown in Figure 1.1.

Simulation is an example of a scientific instrument that extends human abilities. In moving down the spectrum from extrapolation to augmentation, the link to reality (which we refer to as the *domain*) becomes more tenuous or abstract. Equally, the (implied or explicit) *model* of reality plays an increasingly important role in understanding the outputs of the scientific instrument.

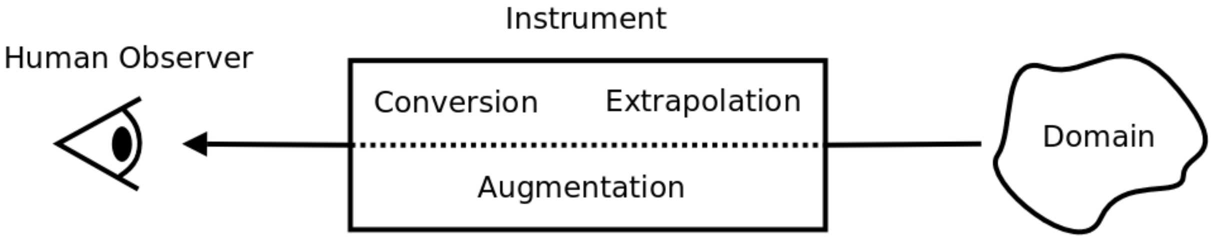

Figure 1.1: Illustration of how scientific instruments enhance the scientist's innate abilities. Arrow denotes an observation.

An important relationship between an instrument and the thing it attempts to measure is the inputs which the instrument receives. An instrument such as a microscope takes a direct physical input and extrapolates accordingly. Again, the relationship becomes more complicated down the spectrum. For an NMR machine, there is a property of the real domain that is under constant measurement, but this is interpreted via a mathematical model. For simulation, the properties of the real domain are often estimated or abstracted: the environment, the input parameters, the number of layers and the forms of interaction are necessarily simplified. This puts a significant responsibility on the modelling – without a principled approach to development, the results of a simulation cannot be interpreted in any meaningful way.

1.4.1 Does it look right?

People are often unwilling to accept the output of instruments. Some of the reasons for this are:

1. They do not know how to use the technology;

2. They do not understand how the technology works;

3. They do understand how the technology works but do not believe the instrument is doing the right thing, i.e. they think that it is built on incorrect abstractions;

4. They do not appreciate how the outputs of the instrument relate back to the domain (reality): a technical novice may accept, use, or engage with an instrument if it looks like the system it is simulating; inhibitors to use include working at the wrong level of abstraction for the domain experts and failure to use domain language;

5. The research culture as a whole does not accept the use of the instrument, i.e. it is not a standard technique, and others have some of the above reservations.

As people who develop such instruments, we need to be aware of the above, and make these instruments as accessible as possible without trying to deceive (i.e. do not make it look like the system if that is not in fact how it is working). The CoSMoS process is addressing the issue of trust in science and simulation for science by seeking to support the collection of evidence and the presentation of validation (see [10]).

Chapter 2

CoSMoS Process Concepts

This chapter outlines the main elements of the CoSMoS process, in terms of phases, products, activities and roles. This defines the terminology of CoSMoS development. The ways in which these concepts can be used in engineering simulations is the subject of chapter 3.

The CoSMoS process provides a general structure within which complex systems simulations can be developed and used. In the introductory chapter we highlighted that this is necessarily an interdisciplinary process that captures a way for scientific discovery to be conducted with the aid of computer simulation. The process therefore supports open-ended scientific research that incorporates the relationship between science (what questions are we addressing) and engineering (the construction of a simulation tool to address the scientific question). It is never the case that a team of developers hands over a tool to a team of scientist clients; both are inherently part of the process from inception to conclusion.

An application of the CoSMoS process is referred to as a *CoSMoS project*, or simply a *project*. Recall that the aim is to support a series of simulations, not a one-off throwaway simulation. Thus, a project involves the construction of a simulation platform, which supports instances of the simulation needed for simulated experiments.

The process concepts are described without reference to any specific tools or technologies; the goal is to express the generic elements and driving forces behind the construction and use of a CoSMoS style simulation. The diagrams in this section are akin to "lifecycle" models: they show concepts and connectivity, but do **not** show prescribed routes or dependencies.

2.1 Overview

The three key structuring concepts of the process are *phases*, *products* and *activities*. The description of phases relates to the different stages required for engineering a simulation platform to be used for scientific research. Throughout the phases, interrelated products are created and modified. These products provide a way of structuring, capturing and documenting relevant research and technical outputs, and support reproducibility by understanding how the outputs relate to aspects of their construction. The way in which products are updated is driven by activities. In this section, the phases, products and activities are summarised. The following sections elaborate on these features.

The three phases of the CoSMoS process have been identified from a series of case-studies (see [10]; an extended description of the case-studies will appear in subsequent documentation). The purpose of phases is to capture distinctly different motivations, which structure the way other process concepts are considered. The phases are:

Discovery Phase: establishes the scientific basis of the CoSMoS project; identifies the domain of interest, models the domain, and elucidates scientific questions. The phase is concerned with science, not simulation.

Development Phase: produces a simulation platform to perform repeated simulation, based on the output of discovery.

Exploration Phase: uses the simulation platform resulting from development to explore the scientific questions established during discovery.

A CoSMoS project naturally begins with a discovery phase followed by development and then exploration. However, the challenges of understanding and mimicking complex behaviours means that CoSMoS does not attempt to constrain iteration and alternation among phases.

The CoSMoS process identifies products that represent artefacts created and modified during a project. Whilst these artefacts may be tangible or physical in many situations, in low-criticality or low-impact projects, any or all artefacts (except the actual simulations) may be implicit. The five CoSMoS products are:

Research Context: captures the overall scientific research context of the CoSMoS project. This includes the motivation for doing the research, the questions to be addressed by the simulation platform, and requirements for validation and evaluation.

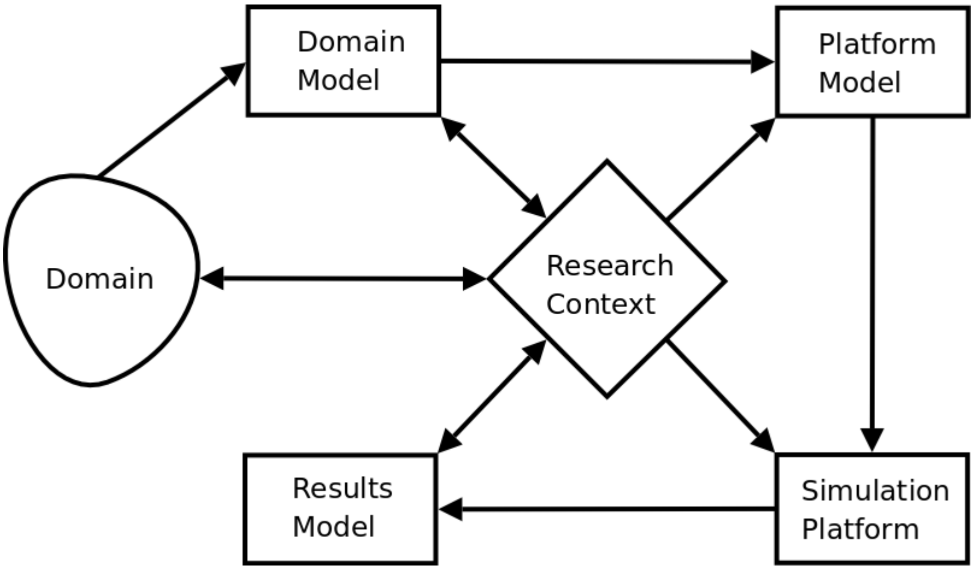

Figure 2.1: Relationships among products (rectangles) and the domain of interest. Arrows give a sense of the information flow involved in development of products. The Research Context is the central and unavoidable product, on which all other products are dependent

Domain Model: encapsulates understanding of appropriate aspects of the domain into explicit domain understanding. The domain model focuses on the scientific understanding; no simulation implementation details are considered.

Platform Model: comprises design and implementation models for the simulation platform, based on the domain model and research context.

Simulation Platform : encodes the platform model into a software and hardware platform upon which simulations can be performed.

Results Model : encapsulates the understanding that results from simulation: the simulation platform behaviour, results of data collection and observations of simulation runs. Note that the way that the domain model captures the relevant understanding of the domain is mirrored by the way that the results model captures understanding of the simulation platform.

Figure 2.1 summarises the relationships among the CoSMoS products, showing how information flows between the products.

The CoSMoS process iterates through the three phases until a stopping condition is met. At each stage, the products may be updated, though it is also possible to simply

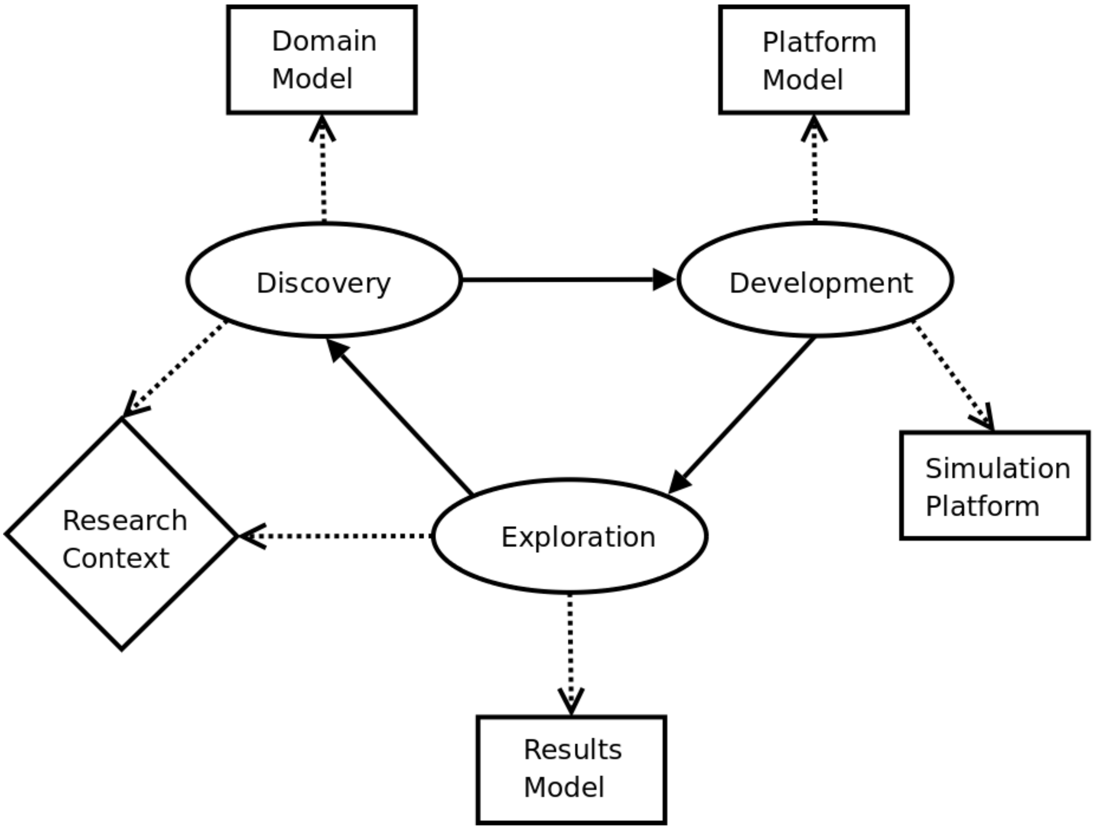

Figure 2.2: Idealised relationship between products and phases. Solid arrows denoted the progress of a project through phases. Broken arrows indicate the primary product focus of each phase

review the products and pass on to the next stage. Considering the different motivations of each phase, they each focus naturally on the modification of a subset of the five products. The typical route would be: the discovery phase develops the domain model and establishes the scientific aspects of the research context; the development phase focuses on building the platform model and simulation platform; the exploration phase creates the results model and adjusts the research context accordingly. Figure 2.2 expresses these relationships between products and phases.

By iterating through phases, we should achieve systematic development that aids scientific reproducibility, and supports subsequent modification and interpretation. The first pass initiates development of each product. Later iterations may modify, extend, or merely represent cursory revisitings of, products. A common situation in the case studies (see [10]) is that the exploration phase generates a results model that points to missing concepts in the domain model. The next pass's discovery phase would then address the scientific import of the change (is it, or can it be, scientifically justified? Do we need to investigate more background?) and necessary changes can be incorporated into the domain model. It may be the case that no action needs to be taken in a specific phase; for example, later development phases may not require modifications to the simulation platform, if a new set of experiments can be performed by simply adjusting parameters.

Whilst figure 2.2 presents the relationship between the products and phases, it does not preclude modifying products not naturally associated with the motivation for the current phase. For example, one might build a prototype simulation platform (updating all the relevant products) during the discovery phase, to elucidate elements of domain knowledge and to guide the construction of the domain model and research context products.

Within the phases, the five products are modified. Creation and modification are undertaken as *activities* in the CoSMoS process. Activities describe the high-level things that we do to update aspects of the process products. Five generic activities are identified that apply to each of the three phases:

1. Scoping
2. Modelling
3. Experimenting
4. Documenting
5. Interacting

In common with other development processes (for example the Rational Unified Process [7]), the CoSMoS process activities are associated with *roles*. The purpose of this association is to clarify the nature of the activity – like other engineering methods, the roles have many-to-many-optional mappings to the people participating in a project. Roles are related to responsibilities within a project, for example, responsibility for information about the domain.

The following sections elaborate phases, products, activities, and roles. However, before moving on, it is important to remember that there is more to simulation than getting the active components right. At the start of a discovery phase of an agent-based simulation project, the focus is usually on working out what will be represented as an agent – abstraction levels, variation among agents, quantity of agents etc. Agent components are the natural focus of scientific researchers and most simulation. However, a simulation has to model the *environment* as well as the agents, and much effort is also needed in understanding the environment, and abstracting to an appropriate representation. Normally, the environment is inherent in the scientific domain, but, for simulation, the development phase must build a simulation of the environment as well as of the agents.

It has been suggested that describing the activities and products of the CoSMoS process in terms of models tends to focus attention on the engineers or developers of the simulation. This under-states the purpose of the products and their development. Whilst the scientists are expected to do the science, and the developers to do the engineering, in all phases, and in relation to all products, the CoSMoS process requires the mutual involvement of those whose primary roles are in the domain (scientists etc), and those whose primary roles are in simulation engineering. This involvement is often simply that each set of roles knows what the other is doing. The process represents a process of mutual learning and mutual trust. The issues and challenges of this are considered in more detail in [10].

2.2 Phases

The purpose of defining phases is to separate the major concerns of a CoSMoS project. Different approaches are used in each phase, with different associated human input. In a scientific development, the discovery phase is about scientists identifying a scope and issues, and about developers learning from scientists. In the development phase the developer uses appropriate engineering methods and techniques to create a simulation platform; the developer expertise is prime, but the scientists have to validate what is being implemented, and learn from the developers. In the exploration phase, the roles come together in understanding the simulation behaviours and results, in the context of both the science and the engineering.

2.2.1 Discovery Phase

The driving force behind the discovery phase is the need to understand and scope the research that is going to be conducted on the engineered simulation platform. The goals, in relation to scientific research simulation, are:

- To identify the scientific basis for a CoSMoS project, establishing the domain of interest;

- To understand the domain of interest and capture a model of this understanding;
- To establish a set of questions to ask of the domain model via simulation.

Given these aims, the discovery phase result in the creation and modification of the domain model and research context products.

The definition of the discovery phase refers to a *domain* and a *domain model*. The distinction is important if the scientific research context is to be properly understood. The domain is defined here as the general area of study – a scientific specialty (such as auto-immune disease) or an engineering context (such as robotics-swarm organisms). Where the roles of domain expert (client, scientist etc) and developer (engineer, supervisor, etc) are distinct, the domain is the concern of the domain expert. It is the domain expert's responsibility to express what might be relevant to the simulation developer. The developers' job is to clarify what they have been told and interpret it into a domain model that becomes the basis for developing the simulation (section 2.3.2).

Through interaction with the domain expert, a developer gains understanding of part of the domain. In later phases, this understanding can feed into suggested simulation experiments, which should then be discussed and agreed with the domain expert, so that the research context is mutually understood. The distinction of domain and domain model, and the distinct roles associated with them, is important for the research context and the results model, used to interpret simulation results into scientific findings. Science is, and remains, the responsibility of the domain expert.

2.2.2 Development Phase

The purpose of the development phase is to engineer a simulation platform upon which to carry out the scientific research identified in the discovery phase. The development phase encompasses two aims:

- To transform domain understanding into an implementable model of a simulation platform that can be used to undertake the research identified in the discovery phase;
- To develop the model into an implementation of appropriate quality, flexibility and reliability.

We thus have two products upon which most of the tasks are performed: the platform model and the simulation platform. Depending on the criticality and impact of the intended use of the simulation results, the development phase may be agile, lightweight, or a rigorous engineering exercise. In most cases the intended use of simulation results is to explore a scientific question as part of a wider set of scientific research goals and research activities.

At the end of the development phase, the simulation platform exists, and should be in a form suited to the research objectives identified in the discovery phase. The developer should be able to give suitable undertakings on the quality of the engineering – appropriate unit testing and other quality-enhancing activities should have been undertaken. This separates the engineering aspects of validation[1] from the scientific aspects of validation, which relate to whether the simulation has anything meaningful to say about the domain.

2.2.3 Exploration Phase

The exploration phase uses the simulation platform to address the scientific research identified during discovery. This is where the relevance of the simulation results has to be addressed. To do this, and to complete the research objectives that led to simulation, a number of specific simulations may be derived from the platform (with or without the need to change the platform model or research context). Exploration applies both to the generic simulation platform, and to the specific simulations. The phase can be expressed as:

- Performing experimentation on the simulation platform to generate results;
- Assessing outputs and analysing results: evaluation, scientific validation etc.

There are two sorts of output from the exploration phase: discoveries about the simulation and discoveries from the simulation. The former relate to the adequacy of the results; the latter contribute to the research domain.

Discoveries *about* the simulation relate to scientific validation activities of the exploration phase and determine whether the simulation produces qualitatively similar results from qualitatively similar behaviours (as explored by, for instance, [17]). Even if the simulation is a good engineering product (the assumed endpoint of the development phase), mismatches have many possible causes:

- The development phase produced a simulation platform that does not adequately capture the behaviours or interactions of the agents and/or the environment.

[1] The relationship between complex system engineering and conventional systems engineering activities is discussed by a range of researchers within and beyond CoSMoS, and is not elaborated here: see for example [11, 12, 3].

This points to issues relating to realisation of the domain model: the simulation may be sound, but the design decisions or assumptions are at fault.

- In the discovery phase, the domain model contained misunderstandings or inappropriate abstraction from the domain. This focuses on issues relating to the assumptions and decisions taken in arriving at a mutually-understood and sufficient domain model. The problems could relate to the understanding of active components of the system, to understanding of the environment, and/or to understanding of interactions. Furthermore, there are scale issues: failure to produce expected behaviours may result from the wrong (absolute or relative) numbers or types of agents; the wrong relative sizes of components, and many other possible failures or omissions. This can be summarised succinctly but unhelpfully as a failure to include in the simulation the essence of the required emergent behaviours.

- In the underlying science (the domain), there may be fundamental misunderstandings. In this case, the simulation may be doing exactly what is intended and designed to do, but the science (as presented by the domain expert) is wrong.

The interpretation problem is compounded by the likelihood of more than one sort and instance of these causes. In the example studies considered so far (see [10]), issues uncovered during scientific validation often prompt further iterations of the CoSMoS process. The process is similar to that of laboratory science: a continual querying of the evidence and the basis of the evidence, until reasonable certainty is achieved.

Discoveries *from* the simulation are the results. These may be data or visualisations that can be used to complement scientific research, or they may be qualitative understandings of what processes might or might not be involved in certain complex behaviours. The discoveries must always be considered in the context of the domain model, the engineering (development phase products) and the domain.

2.3 Products

The CoSMoS phases and the products are closely interrelated. The products are extended and modified in successive phases and iterations of the process. This section is a descriptive rather than prescriptive characterisation of each product. The aim is to achieve clarity about the CoSMoS process through separation of concerns.

2.3.1 Research Context

The Research Context defines the fundamental scope and purpose of a CoSMoS project. The product may be a diverse collection of references, definitions, understandings and agreements: it collates and tracks the scientific background, the authorities, the technical and human limitations (resources) of the project, and other contextual underpinnings of the project. The research context is "owned" by the domain expert, and relates to the domain concept (section 2.1).

There are two aspects to this product, covering the scientific context, and the requirements of the specific project. In a scientific project, the research context might state high-level goals, research questions, hypotheses, general definitions, success criteria. Requirements of a specific project might include the statements against which the simulation and its outputs can be validated and decisions relating to the level of resourcing available and the level of fidelity needed.

In determining whether a research context is "adequate", consideration should be made of the intended criticality and impact of the simulation project. *Criticality* relates to the role that the simulation project plays in the domain. If the simulation is a speculative exploration of possible factors, it may be low in criticality; however, if the goal of simulation is to provide a missing link in the scientific understanding of a key natural process, then it may have very high criticality. *Impact* is similar, but not identical: a non-critical simulation might have disproportionate impact if it relates to a new or under-researched domain, whilst a high-criticality simulation may have low impact because, once it has identified the critical link, this is confirmed by accepted scientific techniques. If impact or criticality is judged to be high, then the research context must explore how the validity of the engineering (development phase) and the science (exploration phase) are to be addressed, and what will be acceptable. It would be expected that some formal record or argument of validity be constructed, that makes clear the assumptions and limitations of the science, the engineering, and the relevant decision making processes. However, where criticality and impact is low, the recording, structuring and evaluating of evidence and assumptions can take a much more implicit place in the process.

2.3.2 Domain Model

The domain model is a model of the agreed scientific basis for the development of a simulation platform. It can be considered to be an explicit description or a mental model, but if the project is remotely critical, an explicit domain model is needed, to support interaction and mutual understanding.

The term *model* applies to any abstract description: text, informal sketches, diagrams with defined syntax and/or semantics, mathematics, programs, or whatever is chosen. A domain model may be a collection of informal models of the relevant parts of the domain (it is important to record where these come from, so that later misunderstandings can be followed up), or it may be a rigorous, perhaps even an executable, model of known concepts.

In science, models are often presented top-down, as a reductionist separating out of the relevant concepts and behaviours: this can be seen, for example, in the pictures used by biologists to present systems such as the immune, neural and endocrine systems. This is a good starting point for domain modelling, which is then worked on jointly by the scientist and developer. As noted in relation to phases (section 2.2), an essential feature of the domain model is that it includes both the relevant agents and the *environment* in which they interact. Both are subject to abstraction, and both entail assumptions and scientific decisions as to what is relevant.

The domain model is a model based on the science, the domain. It should be free from implementation bias, in that the model should be accessible to the domain expert, and free of concepts related to simulation platform construction[2]. However, the level of detail and abstraction in the domain model depends on the goals and the type of simulator that is needed, determined in the discovery phase (possibly through several iterations of the CoSMoS process) and captured as part of the Research Context product. Most simulation projects focus on two or three *levels* of a complex system, and abstract other levels to their effects on the levels-of-focus; this focusing and abstraction takes place in the context of developing the domain model. For example, we may wish our simulator to operate at the level of cells, thus we may not have to consider the specifics of bio-chemical receptor interactions in the domain model, but we do need to ensure that the abstraction from biochemistry is adequate for the purpose of the simulation.

[2]An interesting situation arises where an implementation model is integral to the science: for instance, in ecological abundance research, a regular grid is as much a feature of the domain as it is of the implementation, owing to traditional sampling and modelling approaches.

2.3.3 Platform Model

The concept of a platform model corresponds, at least loosely, to specification and design models used in conventional systems engineering. The platform model is an engineering derivation from the domain model, and a step towards simulation construction. As in all the products of the CoSMoS process, there could be a formal suite of design models, or there could be a simple mapping of concepts in the domain model to implementable concepts in a simulation medium (some of the ways of doing this are explored in related publications [14, 1, 12]).

The goal of the platform model is to detail the implementation behaviour and interactions of the agents and environment from the domain model. The platform model focuses on the "how" aspects of the domain model: the behaviours that are going to be implemented and executed in the simulation platform.

Compared to the domain model, the platform model should explicitly remove any features that are required to be emergent properties of the simulation platform (the high-level domain behaviours): these are things against which to validate, not implementation concepts. Instead the platform model is a design for the local behaviours that are hypothesised to be responsible for the desired emergent properties.

The platform model narrows down the possible target technologies for the simulation, and addresses the constraints and opportunities of those potential technologies. This might dictate that some concepts in the domain model are further abstracted or simplified, to allow efficient implementation. The engineering design decisions are part of the product, and are needed in the exploration phase to assess the outcome of simulation.

To be adequate for purpose, the platform model must be able to support the engineering requirements of the project. One of the key things that the platform model adds to the domain model concepts is instrumentation to allow observation (visualisation), interaction, and recording of the eventual results of using the simulation platform. Amongst other things, the platform model may also add implementation-related detail such as data structures, control features, and algorithms. The model must support testability and other verification activities appropriate to the criticality and impact of the project.

2.3.4 Simulation Platform

The simulation platform is the generic result of the development phase. It is the basis for specific simulation activities, in the exploration phase. It is a realisation of the platform model, upon which simulations can be performed.

The form of the simulation platform is dependent on the simulation technologies and purpose. It may comprise hardware and/or software. The simulation platform includes the default settings for the architecture and parameters, and realises the default input-output mechanisms (how to put data in, get data out, change parameters, etc). If the simulation platform is considered complete, then key engineering activities (including verification and testing of the implementation) should have been completed. (However, scientific validation is part of the exploration phase, which uses the simulation platform.)

2.3.5 Results Model

The results model is the product that represents and accumulates the output (literal and otherwise) of the exploration phase. In the same way that the domain model is an understanding of the domain being investigated or explored by the project, the results model is used to establish an understanding of the simulation outcomes. Like the domain model, it can take many different forms, and have many different sorts of model within it.

The results model is constructed by experimentation and observation of simulations. Thus, it can record observations, screen-shots, dynamic sequences, raw output data, result statistics, as well as qualitative or subjective observations. Ideally, the model should capture all that we know about the behaviour exhibited by the simulation, at the same level of abstraction as the domain model.

The results model is the basis for interpretation of what the simulation shows. Once constructed, the results model must be considered in the context of the domain model and the research context (see exploration phase, section 2.2). Given the iterative nature of simulation-based research, and the CoSMoS process, the exploration phase could also lead to modification of the research context product and/or the domain model.

2.4 Activities

The CoSMoS process identifies five generic activities that apply to each of the three phases. Activities describe the high-level things that we do to update the products. Activities are carried out by roles (section 2.5), and can modify any number of products, in any number of phases. As with the other concepts of the CoSMoS process, the activities are not prescriptive, and the extent and rigour of their execution depends on the criticality and impact of the project.

The five activities (scoping, modelling, experimenting, documenting, and interacting) were arrived at by analysing the processes used in CoSMoS case studies (reviewed in [10]).

2.4.1 Scoping

Scoping is an activity that aims to circumscribe the work or content of a phase or product. It can relate to the scientific and technical outcomes of the project as a whole. Typically, therefore, this translates to modifications to the research context product. However, the development phase, with its engineering design decisions, also includes significant scoping activities.

Scoping needs to take account of resourcing, scientific objectives, the state of knowledge of the domain, levels of expertise. Scoping also identifies what level or risk or criticality might pertain to the simulation project. The activity establishes the boundaries of the system, the treatment of the environment, and the levels included in the simulation platform. Determining the granularity of a simulation, its agents, environment and anticipated emergent behaviours, is also part of the scoping activity. Scoping also helps to determine, in the research context product, the criticality and impact of the project.

2.4.2 Modelling

A modelling activity is anything that produces a model/description, with the goal of recording or communicating understanding. This could be aimed at communication among roles, or at recording decisions and assumptions, or at exploring through abstract representations. In general, modelling aims to reduce ambiguity – though the extent to which this is achievable is bound up with modelling languages and qualities, and covers several research fields of its own.

In the CoSMoS process, modelling activities always result in a modelling artefact – although we have seen that the products that propose modelling artefacts allow as loose an interpretation of model as is appropriate to the project and phase concerned. In relation to the development phase, Polack et al [10, 12] summarise some of the ways in which engineering approaches taken to modelling have been used in CoSMoS studies.

A novel aspect of CoSMoS modelling is its proposal for validity arguments. Essentially, the validity of simulations is represented as an argument structure (a model), that exposes to critical review the evidence of validity: the assumptions, design decisions, results and so on that lead those involved to accept the scientific outputs of simulations. The arguments also define the limits of their acceptance, the scope within which the simulation outputs are trusted.

2.4.3 Experimenting

The goal of experimenting activities is to extract understanding from a system/entity, and to generate supporting data or evidence.

In common with other meanings of experimentation, an experiment perturbs a system and records its behaviour. Experimentation can be used in any of the CoSMoS phases, and supports a highly iterative approach to simulation development.

The result of experimentation can feed back into any CoSMoS product (e.g. simulation platform, results model). Experimentation also informs the scientific evaluation of the simulations – and, indeed, the engineering verification activities of the development phase are experimentation activities on the implementation.

2.4.4 Documenting

Documenting is a controversial part of any development activity: much time can be wasted in superfluous documentation. However, CoSMoS has designated documentation as a core activity because of the complexity of not only the simulations and the domains, but also of the process of achieving a worthwhile outcome.

The purpose of documenting is to record things, to make them explicit. Documenting is associated with all products and captures important concepts such as abstractions, design decisions, and assumptions. Depending on the criticality and impact of the project, it may be sufficient to use lab-book recording (this is probably the minimal acceptable documentation, as it is conventional in scientific research), or it may be necessary to set up structured recording of sources, decisions, and assumptions[3].

A useful guideline is to consider what level of documentation would be needed for the appropriate level of confidence in the simulation (where appropriateness relates to the criticality and impact: are we convincing ourselves that a particular simulation is valid, or the whole scientific community).

2.4.5 Interacting

Interaction activities among the roles involved in a project are critical in all the simulation contexts considered by CoSMoS. The importance of collaboration and interaction is spelt out in [10], which also describes how suitable interaction structures were established in various CoSMoS studies. An essential element of the discovery phase is determining the bases for interaction.

The goal of interacting is to share and assess the products and activities of the CoSMoS process. Interaction both establishes and critically reviews the current state of phases, and signals when to move forward to the next phase.

Interaction is fundamentally tied to the roles played in the CoSMoS process. However, it is also about the willingness of participants to engage in interaction, and in exposure to another discipline (science; engineering).

2.4.6 Phase-specific activities

In addition to the generic activities, there are some phase-specific activities. In these activities, there is a specific goal that is directly related to the aim or motivation of the phase, and the activity is inherent to that phase. One example of a phase-specific activity is outlined later (section 3.2.1).

2.5 Roles

We have already discussed the need, however artificial, to introduce roles in order to consider how activities happen and products are created in a project: the need is common to most descriptions of engineering processes. Roles can be used to show how collaboration occurs in achieving goals, performing activities, and generating prod-

[3]The need to capture and structure evidence for evaluation purposes is the focus of ongoing investigation and the subject of further CoSMoS-related research proposals.

ucts. Any number of roles can be involved in an activity. Roles do not necessarily map one-to-one to people.

At this stage, the identification of roles associated with the CoSMoS process is on an at-need basis, and further work is needed on clarifying where roles are really separate or relevant. The following is an incomplete list and discussion of candidate roles.

- Platform Modeller, Platform Implementer (aka Developer): the roles of platform modeller and platform implementer relate directly to the products of the CoSMoS process; the role of developer relates to the overall engineering in the development phase, the engineering practitioner. Breaking down the developer role will most likely correspond to roles identified in other development processes (for example agile methods such as eXtreme programming, and feature-driven development).

- Domain Modeller: the role that is responsible for the domain model, that is, for determining the understanding of the domain that is needed for the simulation, and for representing that understanding in whatever way is appropriate[4].

- Documenter: a role with responsibility for the documenting activity, that is, for recording whatever is deemed necessary to the project (as in activity description) at what ever level is deemed appropriate.

- Domain Expert (aka Scientist): the role that "owns" the science, determines the scope of the domain model, and has the ultimate say on whether the simulations do something credible.

- Scientific Researcher: this role relates to the detailed laboratory research that contributes to the domain and results models, and the provision of detailed information and answers to developers; it is a doing, rather than a controlling, role.

- Client: in the same sense as the client who commissions a computer application, this role can determine that a project will start or end. It may be synonymous with the scientist or the supervisor role in some projects

- Supervisor: a role that stands back from the activities of the CoSMoS process, ensuring that the other roles remain focused on the task in hand.

[4]There might be multiple domain expert roles within on CoSMoS project – for example where a simulation expresses a layered complex system with different competencies associated to different layers, or where separate domain expertise is needed to model aspects of the environment. In this case, the responsibilities of the different domain modeller roles need careful consideration and delimitation.

Chapter 3

Phases and Activities

This chapter considers how the activities might relate to the phases of the CoSMoS process, highlighting some core aspects in greater detail. The aim is to give examples of how phases and activities come together to produce and maintain products, rather than to do an exhaustive cross-correlation. The rationale for this approach is firstly readability, but secondly because the CoSMoS project is ongoing, and because the range of possible applications is potentially unlimited. This means that it does not make sense to try to identify all possible routes through the process.

The chapter is structured by the three phases; within each phase, some of the core activities are identified for consideration. The level of detail and coverage is variable, for the reasons noted above. Under the development phase, a phase-specific activity is included, as an example of this concept.

3.1 Discovery Phase

In this phase the Scoping, Modelling and Documenting activities are chosen for inclusion.

3.1.1 Activity: Scoping

Discovery scoping establishes the purpose and desired outputs of the scientific aspects of the research we are undertaking. Essentially we are scoping the research to be carried out. This involves identifying what we are going to do, why we are doing this, and who we are going to work with. Additionally, we establish what are the precise questions being asked of the domain of interest. As the various products are constructed our understanding of the domain improves and questions we wish to ask of it are elaborated. This activity captures the details of the scientific questions to be asked by updating the research context product.

Candidate roles that might be involved in this activity are the *client*, *domain expert*, *domain modeller* and *supervisor*.

Typical discovery scoping would involve:

Goal identification : explicitly stating what the over-riding goal of the research is. It is probable that specific research questions may not be apparent in the early stages, so the goal may be a high-level statement of the sort of outputs that are desired. The goal should satisfy both the client (it is a sensible and useful piece of scientific research) and the supervisor (it is an achievable piece of work to be modelled and simulated).

Domain of interest identification : establish what is/are the system/s that will be the subject of the research. The domain (along with the domain expert role) will be the source of input and inspiration for the scientific research to be carried out. In many cases the domain is easily identified, however, in other cases an investigation into different possible domains may be needed before a suitable domain is identified. This choice of domain may be affected by the goal of the research and/or the people fulfilling the different roles: choice of domain might well be biased by the client and supervisor's knowledge (or lack thereof).

Criticality of the project : what is the impact of the work, what is its feasibility and its achievability? This might involve deciding the level of argumentation required, and if so what type of argument this should be. In many cases an argument regarding technical and scientific work is either left implicit or constructed *ad hoc* for research reports/publication. In some cases, the argument needs to be made explicit. We need to establish the need for such argumentation as early as possible in the work so that the required evidence can be captured and structured during all subsequent activities[1].

We also need to identify to what level of detail we are going to examine the domain. An identified goal should give us insight into the level of abstraction and granularity we are going to model and simulate the domain. This level of abstraction might include the (agent) granularity of the modelling/simulation. Are

[1] It is possible that the initial arguments would have to be refactored or revised as the simulation developed, and as understanding develops

we interested in generic properties or do we need detailed representations of domain concepts? This is often determined by the time and cost available to the CoSMoS project.

Establish desired domain properties : identify the desired properties of the domain model that we wish to capture in the simulation platform. In addition, identify how they could be measured in a scientific context, e.g. is any emergent property recognised or measured; qualitative vs quantitative measurements.

Establish limitations : what are the limitations that the domain imposes on a simulation platform? Typically this would include any scaling requirements of a simulation in order to be useful from a scientific point of view. For example this would consider the minimum numbers of agents that are needed.

Identify success criteria : based on the scientific basis of the CoSMoS project, success criteria can be identified that state what a successful project would achieve. These criteria can be used when evaluating the work, and relate to the successful generation of new knowledge from the simulation, or meeting a particular specification.

3.1.2 Activity: Modelling

In the discovery phase, the main modelling activity relates to the construction of the domain model product, a representation of domain understanding. The roles involved in domain modelling are the domain modeller and domain expert.

There are many elements to the activity of domain modelling, which can include:

Acquiring domain background knowledge : before constructing the domain model, the domain modeller may need to acquire a level of background knowledge to enable them to better appreciate the domain to be modelled and hence choose an appropriate modelling approach. Typically a domain expert will be able to provide the domain modeller with access to suitable domain knowledge in the form of verbal communication, text-books, manuals, research articles, etc. The level of background knowledge that will be required should become apparent through domain modeller and domain expert interaction. Much of the background knowledge will not form part of any domain model product, but simply enables the model construction to take place. The domain modeller should document domain background as they see fit.

The domain expert (the scientist advising the project) must also get used to the level and type of detail that is required for the construction of the simulation platform. There is often a large amount of knowledge about a domain that is not relevant to the required domain model, but where knowledge is relevant, the domain expert needs to present this knowledge in a form accessible to the domain modeller.

Identify domain modelling tools : the domain model product is more often than not an explicit model, hence a decision should be made to how the domain model is going to be captured. Domain modelling tools can take any suitable form (determined by the domain modeller, as appropriate to the goal of a CoSMoS project). In the simplest form this may be textual descriptions or tabulations, or may be a structured modelling language such as a variant of existing modelling approaches such as UML, SBML.

Establish domain behaviours : given the scientific goal, the system-wide behaviours of the domain need to be established. These are often considered as the emergent behaviours of the domain. They establish a focus for the domain model product and provide a reference point for comparing future simulation outputs. Once the domain behaviours are identified, their hypothesised relationship to the domain elements should be established.

Establish domain elements : Domain behaviours are generated by domain elements: for example populations of agents (physical, chemical, biological, social) that interact with each other and the environment in which they exist. Essential to domain modelling is identifying what these agents are, what their environment is, and how agents interact with each other and with their environment. The categorisation of domain elements into agent or environment will often depend on how we are viewing the system.

Simulation prototyping : elucidate domain understanding using simple prototype simulations. Prototyping activities support the "thinking-through" of elements of the domain model.

3.1.3 Activity: Documenting

Documenting is an essential part of the research context product. Documenting involves capturing decision that have been made that complement any explicit model.

A useful idiom in documenting is to construct a structured argument for the research concurrently with other activities. A *documenter* role could coordinate the documenting activity, interacting with various other roles that will depend on what is being documented.

Typical elements of the discovery documenting activity would include:

State research questions : explicitly state any scientific research questions and hypotheses. These questions must have a suitable representation in the domain model product so that the simulation platform upon which this is based is able to address these questions.

Document assumptions : inherent in the process of modelling, assumptions are made. The capture and documentation of assumptions and decisions is discussed further in [1, 12]. Assumptions are essential to understanding a model (and subsequent simulation) in the context of the wider domain that is being represented. Incorrect or bad assumptions can render results meaningless. By documenting assumptions they are accessible when considering the wider context of the work and point to places where models should be altered. Whilst many assumptions are implicitly made and will be missed, the assumptions being made during each modelling decision should be considered. (These assumptions can feed directly into a structured argument in the research context product.)

State Argument Claim : establish and state the central claim of any argument being constructed. The claim is the thing believed to be true, which will be supported by evidence to be collected throughout the activity.

Tool identification : any structured argument should be express in a structured way, therefore suitable tools and techniques need to be identified.

Evidence gathering : evidence should be gathered to support any argument claims. This will involve continually evaluating other on-going activities for relevant details to the argument being constructed. Once collected, evidence can be populated into the structured argument.

3.2 Development Phase

In this phase, the phase-specific activity of platform implementation is outlined, followed by consideration of interrelated scoping and interaction activities, then modelling and experimentation activities.

3.2.1 Platform Implementation: phase specific activity

This is the phase-specific activity of the development phase. It encompasses the engineering tasks needed to implement the platform model as the simulation platform product. Any suitable software engineering techniques can be used to develop and test the simulation platform, in line with the scoping and modelling activities (below). As noted in relation to this phase, testing in this context relates to verification of the simulation implementation, rather than the scientific evaluation: testing of the simulation-level behaviours is part of the exploration phase.

The platform modeller and platform implementer will interact to perform this activity.

3.2.2 Activity: Scoping and Interacting

Guided by the domain model and research context products, platform scoping identifies the type of platform that can address the research questions identified in the research context.

Roles involved in scientific scoping are the platform modeller, implementer and the supervisor. Input may also come from the domain modeller. This, in turn, dictates that the scoping activities are closely coupled to the interaction activity in the development phase (as, indeed, it is elsewhere in the CoSMoS process).

Like all scoping activities, platform scoping develops the research context product, and may also call on documentation (role and activity).

The scoping activity in the development phase should address the following tasks:

Identify infrastructure domain : what technologies are going be used to construct the simulation platform product. The infrastructure includes identifying suitable implementation hardware, any programming paradigms (for example object orientation), implementation languages, tool-chains, libraries, etc. Any consequences of these choices on the ability to address the research context should be documented. The chosen infrastructure must be a sensible choice to address the research context of the project.

Identify technical limitations : investigate any general technical limitations of the simulation platform. Given the chosen infrastructure domain, are there limits to the computational abilities of the simulation platform, e.g. realistic numbers of agents/processes/objects. How do these compare with the scientific limitations of the research context?

Develop specification/s : describe what the simulation platform should be able to do. Identify the relevant inputs and outputs of the simulation platform, and how the user is able to interact with the platform. Examples include the type and format of output data required. The specification may be either formally or informally stated depending on the specifics of the project.

Identify modelling tools : what tools will be used to describe the platform model product. The choice may be informed by both the way the domain model is expressed and the infrastructure domain chosen for the simulation platform.

3.2.3 Activity: Modelling

The platform modelling activity of the development phase translates the domain model into a platform model product that can be implemented to produce a simulation platform. This translation applies an implementation bias (established in platform scoping) to the domain model in order to produce the platform model.

The platform modeller and domain modeller roles interact to perform this activity. Platform modelling will involve some of the following tasks:

Map domain elements : map each element of the domain model into the platform model. This may be a simple one-to-one mapping, but does not have to be. The chosen implementation technologies will affect this mapping

Remove domain behaviours : an important step is to ensure that we do not explicitly encode the domain behaviours that we wish to emerge from the simulation platform. These emergent behaviours should be part of the domain model, but must be removed from the platform model. Platform model elements should be identified as either agents or environment.

Add instrumentation : augment the model with elements to fulfil any specification requirements to visualise and interact with the simulation platform. Typically this activity requires design of how data is input to and output by the simulation, and how the user can interact with the simulation, e.g. a GUI.

Identify assumptions : document any assumptions made when constructing the platform model. Many of these will relate to how closely the domain model elements match the platform model elements, and whether any other additions affect the underlying domain elements and hence the scientific outputs of the simulation platform.

3.2.4 Activity: Experimenting (for calibration)

Platform calibration is an experimentation activity that establishes the relationship between the simulation platform and the domain model it implements. It naturally follows the engineering testing of the platform. The goal of the activity is to understand the link between the simulation outputs and the domain under study, so that simulation instances can be used for scientifically-relevant experimentation in the exploration phase.

Calibration is achieved via a series of experiments with the simulation platform that map known domain behaviours and variables in order to work out the relationship between the variables of the simulator and the variables observed in the domain.

Calibration should be performed by the platform implementer and domain expert. In this respect, it differs (fundamentally) from the verification-related activities of the development phase (above).

Calibration performs the same experimental loop as conventional experimentation activities, concentrating on two main tasks:

Visual calibration : develop a sensible base-line for the simulator that produces desired domain behaviours that the domain expert is happy with, using variables whose relationship to the underlying domain is understood.

Sensitivity analysis : a fuller exploration of the simulation platform's variables and behaviours.

3.3 Exploration Phase

The exploration phase does not currently report any specific activities. In CoSMoS, we are exploring how experimentation takes place with a simulation platform, how the platform evolves to support new experiments, and how the whole experimentation activity relates to the products and phases outlined here. Read et al [14] address the problem of hypothesis derivation, for example, whilst [4] considers experimentation activities related to parameter setting.

Chapter 4

Concluding comments

The CoSMoS project is ongoing, so this document represents the state of the CoSMoS process in March 2010. As indicated in the text, there are a number of published papers that express parts of the process, the philosophy and motivation of the CoSMoS process. This technical report pulls together the existing work, defines terminology, and outlines the concepts of the process.

Further work, and potential further technical reports, include:

1. Completion of the definition of process concepts (including the addition of more optional concepts);

2. Recording and publishing CoSMoS patterns: these relate to all phases and activities;

3. Summarising the examples and case studies that have given rise to, or are using, the CoSMoS process, and using this summary to extend the coverage of how CoSMoS can be used (chapter 3).

References

[1] P. S. Andrews, F. Polack, A. T. Sampson, J. Timmis, L. Scott, and M. Coles. Simulating biology: towards understanding what the simulation shows. In *1st Workshop on Complex Systems Modelling and Simulation*, pages 93–123. Luniver Press, 2008.

[2] M. Calder, S. Gilmore, J. Hillston, and V. Vyshemirsky. Formal methods for biochemical signalling pathways. In *Formal Methods: State of the Art and New Directions*. Springer, 2008.

[3] S. Efroni, D. Harel, and I. R. Cohen. Emergent dynamics of thymocyte development and lineage determination. *PLoS Computational Biology*, 3(1):0127–0135, 2007.

[4] P. Garnett, S. Stepney, and O. Leyser. Towards an executable model of auxin transport canalisation. In *1st Workshop on Complex Systems Modelling and Simulation*, pages 63–92. Luniver Press, 2008.

[5] P. Humphreys. *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Oxford University Press, New York, 2004.

[6] M. Jackson. *Software requirements & specifications: a lexicon of practice, principles and prejudices*. ACM Press/Addison-Wesley Publishing, 1995.

[7] Philippe Kruchten. *The Rational Unified Process: An Introduction, 2nd Edition*. Addison-Wesley, 2000.

[8] M. Mokhtar, R. Bi, J. Timmis, and A. M. Tyrrell. A modified dendritic cell algorithm for on-line error detection in robotic systems. In *IEEE Congress on Evolutionary Computation (CEC 2009)*, pages 2055–2062. IEEE Press, 2009.

[9] N. D. Owens, J. Timmis, A. J. Greensted, and A. M. Tyrrell. On immune inspired homeostasis for electronic systems. In *6th International Conference on Artificial Immune Systems (ICARIS)*, volume 4628 of *LNCS*, pages 216–227. Springer, 2007.

[10] F. A. C. Polack, P. S. Andrews, T. Ghetiu, M. Read, S. Stepney, J. Timmis, and A. T. Sampson. Reflections on the simulation of complex systems for science. In *Fifteenth IEEE International Conference on Engineering of Complex Computer Systems*. IEEE Press, 2010.

[11] F. A. C. Polack, S. Stepney, H. Turner, P. Welch, and F. Barnes. An architecture for modelling emergence in CA-like systems. In *VIIIth European Conference on Artificial Life*, volume 3630 of *LNAI*, pages 433–442. Springer, 2005.

[12] F. A.C. Polack, P. S. Andrews, and A. T. Sampson. The engineering of concurrent simulations of complex systems. In *IEEE Congress on Evolutionary Computation*, pages 217–224. IEEE Press, 2009.

[13] F. A.C. Polack, T. Hoverd, A. T. Sampson, S. Stepney, and J. Timmis. Complex systems models: engineering simulations. In S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, *Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 482–489. MIT Press, Cambridge, MA, 2008.

[14] M. Read, P. S. Andrews, J. Timmis, and V. Kumar. Using uml to model eae and its regulatory network. In *8th International Conference on Artificial Immune Systems*, volume 5666 of *LNCS*. Springer, 2009.

[15] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. BioAmbients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, 2004.

[16] A. Sadot, J. Fisher, D. Barak, Y. Admanit, M. J. Stern, E. J. A. Hubbard, and D. Harel. Towards verified biological models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2007.

[17] M. Wheeler, S. Bullock, E. Di Paolo, J. Noble, M. Bedau, P. Husbands, S. Kirby, and A. Seth. The view from elsewhere: Perspectives on alife modelling. *Artificial Life*, 8(1):87–100, 2002.