Proceedings of the 2015 Workshop on
Complex Systems Modelling and Simulation

CoSMoS 2015

Susan Stepney, Paul S. Andrews
Editors

CoSMoS 2015

Luniver Press
2015

Published by Luniver Press
Frome BA11 6TT United Kingdom

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

CoSMoS 2015

ISBN-10: 1-905986-46-7
ISBN-13: 978-1-905986-46-0

While every attempt is made to ensure that the information in this publication is correct, no liability can be accepted by the authors or publishers for loss, damage or injury caused by any errors in, or omission from, the information given.

Preface

The CoSMoS workshops series has been organised to disseminate best practice in complex systems modelling and simulation, with its genesis in the similarly-named CoSMoS research project, a four year EPSRC funded research project at the Universities of York and Kent in the UK. Funding for the CoSMoS project has now completed, but we have continued to run the workshop series as a forum for research examining all aspects of the modelling and simulation of complex systems. To allow authors the space to describe their systems in depth we put no stringent page limit on the submissions.

We are pleased to be running the eighth CoSMoS workshop as a satellite event at the European Conference on Artificial Life (ECAL 2015), York, UK. ECAL, along with its sister conference series ALife, is the leading international conference on artificially constructed living systems, a highly interdisciplinary research area rich in complexity, which provides a natural complement to the issues addressed by the CoSMoS workshop.

The main session of the workshop is based on five full paper and one extended abstract submissions:

Andrews and Stepney present a small addition to the CoSMoS pattern language: the Domain Experiment Pattern, which provides additional structure for the Domain Model component in cases where the experimental setup and procedure details are an important aspect of the simulation project

De'Bell uses the CoSMoS approach to build the first iteration of a social system hierarchical network model and simulation, targetting community empowerment in the health care domain

Greaves et al. present the details of a project using the CoSMoS approach to build a simulation to investigate a new theory for stem cell decision making; the paper structures its presentation using the CoSMoS pattern language

Hernandez et al. present the 'action horizon', a new tool for analysing and counteracting destabilising events, to aid the control of complex networks, and demonstrate its action by stabilising the well-known chaotic predator-prey system

von Mammen et al. describe how they have integrated the CoSMoS approach in the teaching of their graduate level Interactive Simulation curriculum

Williams et al. provide an extended abstract outlining a Platform Model of a complex immune system signalling pathway, using the X-Machine modelling approach

Our thanks go to all the contributors for their hard work in getting these submissions prepared and revised. All submissions received multiple reviews, and we thank the programme committee for their prompt, extensive and in-depth reviews. We would also like to extend thanks to the organising committee of ECAL 2015 for enabling our workshop to be co-located with this conference. We hope that readers will enjoy this set of papers, and come away with insight on the state of the art, and some understanding of current progress in complex systems modelling and simulation.

Programme Committee

Paul Andrews, University of York, UK
James Bown, University of Abertay, Dundee, UK
Richard Greaves, University of York, UK
Jose Juan Hernandez, University of Las Palmas de Gran Canaria, Spain
Simon Hickinbotham, University of York, UK
Mark Read, University of Sydney, Australia
Susan Stepney, University of York, UK
Danesh Tarapore, University of York, UK
Tim Taylor, University of York, UK
Sebastian von Mammen, University of Augsburg, Germany

Table of Contents

CoSMoS 2015

The CoSMoS Domain Experiment Model

Paul S. Andrews[1,2] and Susan Stepney[1,3]

[1] York Centre for Complex Systems Analysis, University of York, UK
[2] Department of Electronics, University of York, UK
[3] Department of Computer Science, University of York, UK

Abstract. The CoSMoS Domain Model contains information about the real-world system we are attempting to model. The system we are attempting to model and simulate may include details on how that system can be manipulated, probed and analysed and we may wish to capture the experimental setup present in the Domain. Here we describe the relevant Domain Experiment Model, and show its relationship to other core components of the CoSMoS approach.

1 Introduction

CoSMoS provides a structured approach to enable the construction and exploration of simulations for the purpose of scientific research. We capture the real work system of interest as a Domain Model. Depending on the Domain and our Research Context, the system we are attempting to model and simulate may include details on how that system can be manipulated, probed and analysed through real-world experiments. In such cases, we may wish to explicitly represent the experimental setup present in the Domain. This we call the Domain Experiment Model, an explicit component of the Domain Model. Here we describe the Domain Experiment Model, showing its relationship to other core components of the CoSMoS approach.

In section 3 we summarise the basics of the CoSMoS approach, followed by section 3 in which we explore the experimental concepts present in CoSMoS, and identify the need for a Domain Experiment Model. In section 4 we show how the Domain Experiment Model can be incorporated as a component of the Domain Model and how it relates to the Platform Model, and in section 5 we discuss running simulation experiments.

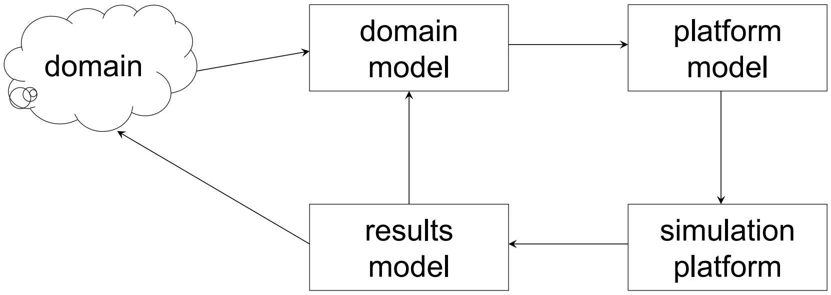


Fig. 1. The CoSMoS core components, which are developed within the Research Context.

2 The CoSMoS Approach

The core CoSMoS philosophy revolves around engineering a properly calibrated simulation platform that suits the criticality and intended impact of the research outcomes. This simulation platform is considered to be a *scientific instrument* [4] and is the basis for running multiple simulation experiments that reveal insight into the modelled domain.

The whole process of simulator construction and use takes place within an overall scientific *research context*. This context identifies the goals and scope of the research being conducted, and includes explicit statements of the simulation purpose, resources, constraints, assumptions, and success criteria. These elements of the research context drive the design, implementation and use of the scientific simulators. Capturing this process is achieved via a series of related *core components* that explicitly describe specific aspects in the building, verifying, and use of the simulator. Together they help provide confidence that simulation results can actually tell us something that relates to the real system being studied.

The CoSMoS core components are explored in detail in [3]. They are shown in figure 1 and summarised here:

Domain: a particular view or perspective of the part of the real world that is the system of study. It describes what the simulation project is “about”.

Domain Model: a model encapsulating the scientific understanding of appropriate aspects of the domain. It provides the agreed scientific basis and assumptions for the development of a simulation platform; simulation implementation details are not considered in this model.

Platform Model: a model providing the high level specification of the simulation platform, comprising design and implementation details, incorporating relevant domain model scientific concepts, research context experimental requirements, and implementation constraints and assumptions.

Simulation Platform: the encoding of the platform model into a calibrated software and hardware platform with which various simulation experiments can be performed.

Results Model: a model that encapsulates the understanding of outputs and results from simulation experiments, in domain terms, enabling comparison with results from domain experiments.

3 CoSMoS and Experiments

Within the CoSMoS approach the concept of an experiment is present at two stages: *domain experiments*, performed on a real world system within the Domain, and *simulation experiments*, carried out within the Simulation Platform. In common terminology, they are analogous to *in vivo*/*in vitro* and *in silico* experimentation respectively.

A common end goal of a CoSMoS-based simulation research is to run simulation experiments on the Simulation Platform that enable us to build a Results Model that can be compared to the Domain Model and provide insight back into the real Domain of study. To achieve this, the Simulation Platform must allow us to run appropriate simulation experiments that reflect the concepts of interest in the Domain. Specifically, simulation experiments should allow us to select the appropriate model components and behaviours, control the initialisation of key parameters, and perform appropriate analyses via suitable statistics. These abilities are explicitly represented within the Platform Model (via *instrumentation*), but ultimately stem from concepts within the Domain and Domain Model.

These simulation experiments are performed to explore and understand the behaviour of the simulation, and to compare that behaviour with assumed or known Domain behaviours. There is a danger, however, that the simulation experiments can be used to explore behaviours that it is infeasible to observe or measure in the Domain, producing incomparable results. So in some cases, depending on the Domain and the research context, it is sensible to ensure that the simulation experiments mirror possible domain experiments, to help ensure comparable results.

The CoSMoS approach already has a specific place to identify and capture simulation experiments, during the construction of the Platform Model from the Domain Model in the Development phase. It also has the Results Model, a description of the behaviours encoded within the

Simulation Platform that are expressed from running simulation experiments. But it has not provided explicit support for capturing domain experimental details along with the rest of the Domain Model during the Discovery phase. The *Domain Experiment Model* is introduced here as a means of doing so.

The Domain Model can be viewed as describing the behaviours present in the Domain that are expressed when probed via domain experiments. The Domain Experiment Model is the place to explicitly model these domain experiments, describing the experimental system present in the Domain, identifying, for example, experimental procedures and protocols, variables and ranges, controls, measurables, data volumes, sample sizes and statistical tests.

CoSMoS now advocates the use of an explicitly defined Domain Experiment Model in cases where the Domain itself includes the experiment system. The Domain Experiment Model structures a Domain Model by:

- capturing how the concepts, structures and behaviours in the domain model are controlled and manipulated.
- detailing what data is collected from the experiments and how that data is then manipulated and interpreted – using statistical methods – to produce the results.

A prime example of where a Domain Experiment Model would be applicable is Aevol[4], from which we have previously reverse engineer a Domain Model [2]. Aevol is an *in silico* experimental *a*rtificial *evol*ution platform [5] in which populations of digital bacteria are subject to Darwinian-style evolution. Its Domain falls within the areas of evolutionary theory and digital genetics focussing on the evolutionary dynamics of the size and organisation of bacterial genomes. The Aevol simulator encapsulates an *in silico* laboratory to test evolutionary scenarios [5], enabling simulation experiments in which populations of artificial organisms evolve within a controlled environment. These experiments mimic those used in real bacterial evolutionary studies, the most famous of which is the Lenski long-term evolutionary experiment [7]. Started over 25 years ago, this experiment has been continually evolving a strain of *E. coli* within a controlled environment. Periodically *E. coli* are removed, analysed and stored, enabling a genomic lineage to be created for future reference and analysis. Aevol provides many of the same tools as this experimental system within its Simulation Platform.

An example of where a Domain Experiment Model is not applicable is described in [6]. In that case the Simulation Purpose is the investigation of a new theoretical model, and the simulation experiments performed in

[4] http://www.aevol.fr/

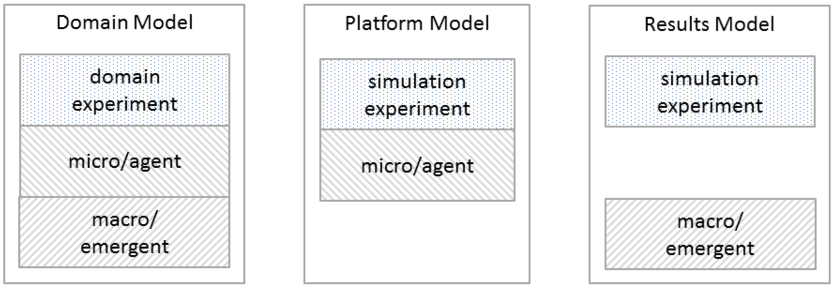


Fig. 2. The substructure of the various CoSMoS models when a Domain Experiment Model is employed.

the investigation are far removed from the particular domain experiments that provide the original data. An explicit Domain Experiment Model would provide no value, and so is not defined.

4 Encapsulating the Domain Experiment Model

The Domain Experiment Model provides extra structure on the Domain Model itself (figure 2); when exploited, this extra structure carries through to the Platform Model and Results Model.

The Domain Model provides the domain concepts and behaviours, and can be factored into three component submodels. The first component is the Domain Experiment Model, which identifies the model parameters and how we manipulate them. The second is the model of the (usually hypothesised) domain micro level structures and behaviours. The third is the model of the domain macro level emergent behaviours.

The Platform Model comprises computational representations of the domain models, making implementation abstractions, and can be factored into two component submodels. The first is the Simulation Experiment Model, derived from the Domain Experiment Model and incorporating instrumentation. The second is the computational realisation of the domain micro structures and behaviours. The deliberate lack of a corresponding macro level model helps ensure that the ‘answer’ – the emergent behaviours resulting from the hypothesised micro level behaviours – is not explicitly coded into the Simulation Platform.

As noted, some Domain Experiment Model concepts are captured in the Platform Model as simulation experiment *instrumentation*, monitoring the core behavioural concepts that are provided by the Domain Model. Concepts in the Domain Experiment Model get translated into Platform Model concepts such as:

- what the explicit parameters of the system are, and how these might be controlled (e.g. fixed constants or variables)
- suitable ranges of operation of the parameters (e.g. sensible parameter ranges or agent numbers)
- termination conditions for experiments (e.g. time condition);
- origins and nature of sources of randomness
- experiment configurator that determines how parameters can change across replicate runs
- what instrumentation is needed to collect, measure and process data from simulation experiments. This includes:
  - visualisers and data loggers
  - Simulation Platform outputs to measure
  - analyses that determine suitable statistical measure
  - analysis conditions that determine the when and how to record and process certain statistics
  - third-party tools, such as Spartan [1]

The Results Model comprises models of the results of running simulation experiments on the simulation platform, and can be factored into two component submodels. The first is a model of the experiment suitable for analysing the simulation data. The second is a model of the domain macro level emergent behaviours in terms of simulation variables. Given that these emergent behaviours are removed from the Platform Model, this model is needed to determine how these behaviours are identified and measured when running simulation experiments.

5 Simulation Experiments

Implementation of the Platform Model that contains the concepts defined in the Domain Experiment Model results in a Simulation Platform capable of running simulation experiments that conform to concepts in the Domain Experiment Model.

The first stage of simulation experimentation is usually used to calibrate the Simulation Platform. This is needed to determine how to *translate* domain parameters and variables into their corresponding platform values (for example how to translate between real-world time, and simulated time), and how to take simulation experiment raw output data and *analyse* it to enable *comparison* with domain results captured in the Domain Model.

Once calibrated the Simulation Platform can be used to run simulation experiments to construct a Results Model. The Results Model can be used *validate* simulation experiment outputs with domain experiment outputs in the Domain Model. If these disagree, it may be because:

- the variables and parameters are not being translated appropriately (calibration may have *overfit* their values)
- there are faults in the Platform Model or in the Simulation Platform implementation (the simulation platform has not been adequately engineered)
- there are faults in the core Domain Model (the science is imperfectly understood), or Domain Experiment Model component (imperfect measurements or statistical errors)

Once validated, a Simulation Platform can be used to run simulation experiments that are analogues of the domain experiments. The raw results of a simulation experiment are analysed and translated into Domain terms, via the Results Model. These analysed results can then be compared to the real world experimental results, and be used to make *predictions* about the results of future domain experiments. Predictions should always be checked against real world data, particularly if the simulation experiment is being run outside the calibration range of the instrument.

6 Conclusions

We have shown how the concept of a Domain Experiment Model can be incorporated into the basic CoSMoS approach, to help structure the various models and the simulation experiments.

Whilst the Domain Experiment Model might not be suitable in all cases, it should be used in circumstances where the Domain knowledge is strongly reliant on a particular experimental system that probes the real-world system under study, and where simulation experiments need to mirror the domain experiments to some degree. It allows for an explicit representation of the experimental system that is present in the Domain, which aids such concepts being adequately incorporated into the Simulation Platform in a more transparent manner. Where this is achieved, simulation experiments (captured by the Results Model) should be directly comparable to domain experiments (captured by the Domain Model), providing improved confidence in simulator outputs.

References

[1] K. Alden, J. Read, M. Timmis, P. S. Andrews, H. Veiga-Fernandes, and M. Coles, "Spartan: A comprehensive tool for understanding uncertainty in simulations of biological systems," *PLoS Comput Biol*, vol. 9, no. 2, p. e1002916, 2013.

[2] P. S. Andrews and S. Stepney, “Using CoSMoS to reverse engineer a domain model for Aevol,” in *2014 CoSMoS workshop*, S. Stepney and P. S. Andrews, Eds. Luniver Press, 2014, pp. 61–79.

[3] P. S. Andrews, S. Stepney, T. Hoverd, F. A. C. Polack, A. T. Sampson, and J. Timmis, “CoSMoS process, models, and metamodels,” in *2011 CoSMoS workshop*, S. Stepney, P. Welch, P. S. Andrews, and C. G. Ritson, Eds. Luniver Press, 2011, pp. 1–13.

[4] P. S. Andrews, S. Stepney, and J. Timmis, “Simulation as a scientific instrument,” in *2012 CoSMoS workshop*, S. Stepney, P. S. Andrews, and M. Read, Eds. Luniver Press, 2012, pp. 1–10.

[5] B. Batut, D. Parsons, S. Fischer, G. Beslon, and C. Knibbe, “*In silico* experimental evolution: a tool to test evolutionary scenarios,” *BMC Bioinformatics*, vol. 14, no. Suppl 15, p. S11, 2013.

[6] R. B. Greaves, S. Dietmann, A. Smith, S. Stepney, and J. D. Halley, “Genome-wide mouse embryonic stem cell regulatory network self-organisation: a big data CoSMoS computational modelling approach,” in *2015 CoSMoS Workshop*, S. Stepney and P. S. Andrews, Eds. Luniver Press, 2015, pp. 31–66.

[7] M. J. Wiser, N. Ribeck, and R. E. Lenski, “Long-term dynamics of adaptation in asexual populations,” *Science*, vol. 342, pp. 1364–1367, 2013.

Towards a Network Model of Community Empowerment for Public Health Outcomes: Application of the CoSMoS Approach to Social System Modelling

Keith De'Bell[1,2]

[1] Department of Mathematics, Statistics and Computer Science,
St. Francis Xavier University, Nova Scotia, Canada. B2G 2X3
[2] York Centre for Complex System Analysis, University of York,
YO10 5DD, UK

Abstract. The role of systematic modelling to construct simulation tools for use in the planning and implementation of community empowerment initiatives is discussed. The CoSMoS approach is used to provide a framework for a demonstration of the construction process, consisting of the development of a simple network model, and to highlight the modelling considerations involved.

1 Introduction

The development of in-silico simulation tools to assist with the planning of interventions intended to improve public health is motivated by two interacting factors. First, it is well established that a significant and sustained improvement in population health necessitates that health inequity and the related social determinants of health be effectively addressed [10, 15, 20, 21]. It has been argued that this requires the coupling of community empowerment processes with more conventional "top-down"public health strategies [17, 18]. In this context, community empowerment is a process in which communities gain more control over those factors that influence their lives including the social determinants of health and includes, for example, improved access to decision making and resources [18, Chapter 2]. Second, the systems within which such interventions occur, i.e. human social systems are inherently complex. The response of such a complex adaptive system to an intervention depends on the context of that specific system. Previous experience with interventions addressing the social determinants of health is extremely important in planning future activities however simple input output models or methods that attempt to remove the specific system context of the intervention are of limited use [14]. Changes in context may even give rise to so-called "wicked problems"; i.e. unanticipated negative consequences that counteract the intended benefits of the intervention. For example, negative impacts of "early intervention"programs in strongly disadvantaged families, even when such programs are considered to "work"in less disadvantaged families, might be considered "wicked problems" [24, and references therein].

In this paper, we describe the construction of a social network model as the basis for investigating the role of social network properties in the context of community empowerment strategies. We limit the scope of the model to those properties that are associated with the building and maintaining of social connections. Despite limiting the model construction in this way, we emphasize that the intention is to highlight the potential for in-silico simulation environments to be incorporated into both the pre-implementation planning and the in-progress adjustment of community empowerment programs. This use of modelling, based on societal structures and dynamics, is, we believe, consistent with the need to combine social science approaches with public health expertise in order to make sustainable advances in addressing health inequities. This combination of social science and humanities knowledge with public health expertise has recently been advocated in a report by the British Academy for the Humanities and Social Sciences [7].

The framework for the model construction is provided by the Complex System Modelling and Simulation (CoSMoS) approach [23]. Designed to enable the development and use of simulation as a scientific instrument, the CoSMoS approach employs a set of modelling stages that facilitate the systematic construction of simulation software. In particular, it has been applied to scientific studies such as the simulation of cellular processes involved in cancer [6]. Its wider applicability has been demonstrated by its use to design algorithms inspired by biological systems [3] and to reverse engineer models from implementations [2]. Recently, its potential utility as a tool for studying social systems has been indicated by its use to formalize and analyze Schelling's (aspatial) Bounded Neighbourhood Model [1] and its use in the context of social-ecological systems [13].

In Sect. 2 the steps of the CoSMoS approach are used to illustrate the construction of the model and some preliminary results are summarized. Sect. 3 identifies a number of modelling considerations necessary for further iteration of the model, and comments on the applicability of simulations based on models of this type to intervention planning.

2 Systematic Modelling using the CoSMoS Approach

2.1 Overview of the Model Building Procedure

Specifying the underlying knowledge base used, and the assumptions and approximations made during the model building process provides an explicit description of the limitations of the resulting simulation package. Thus, these specifications define the class of problems for which the simulation package provides an applicable in-silico environment.

The CoSMoS approach has been described in detail elsewhere [23]. For our present purpose, it is sufficient to visualize it as a set of states (models) with the transitions between states being generated by a set of functions (processes such as abstraction and implementation) (Fig. 1). The specific system the CoSMoS approach is to be applied to, the motivation for the research and questions to be addressed are described in the “Research Context”.

The CoSMoS approach can be used in an iterative fashion. Successive cycles of the iteration are generated in response to an evaluation of experimental results obtained from the simulation. Specifying the research context initiates a process of defining a boundary that determines what is within and what is outside the system to be modelled. This process of setting a boundary is continued in the description of the domain and the construction of the domain model. The risk involved in this necessary process of setting boundaries is similar to that faced by evaluators and intervention planners when drawing boundaries around a system i.e. in modelling a complex system it is not possible to know in advance the importance of these excluded effects in determining the outcomes. Westhorp has discussed these risks in the context of complexity consistent evaluation, and has emphasized the importance of informing the determination of boundaries with theory [24]. In the evaluation of change processes in complex adaptive systems, theory focusses on the components of the system considered important for the change process and, in particular, identifies the interaction set to be included in the evaluation. Systematic modelling enables testing of the effects of exclusions that may be made as part of the modelling process, and an iterative approach to

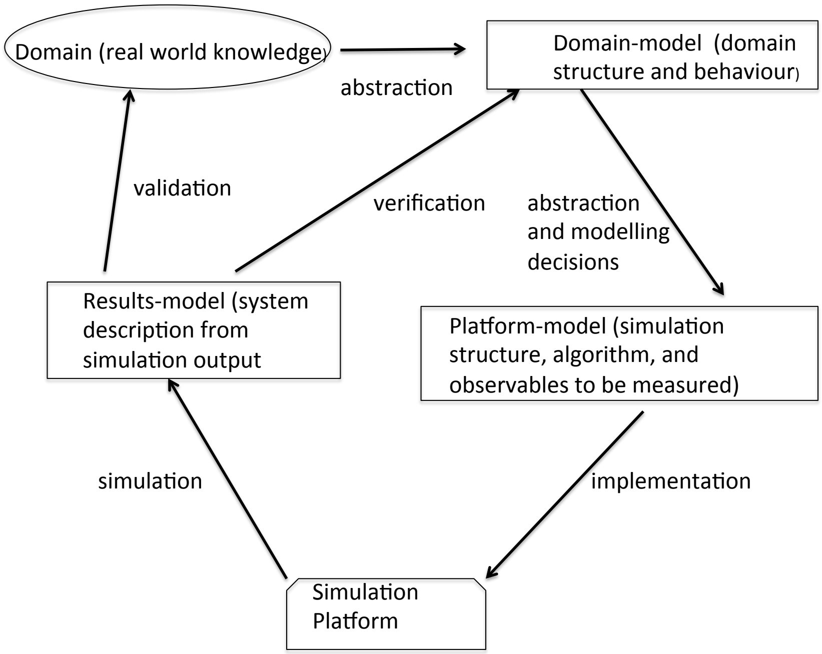

Domain (real world knowledge)
abstraction
Domain-model (domain structure and behaviour)
validation
verification
abstraction and modelling decisions
Results-model (system description from simulation output
Platform-model (simulation structure, algorithm, and observables to be measured)
simulation
implementation
Simulation Platform

Fig. 1. The stages in the CoSMoS approach

refining the model by expanding (or contracting) the system boundary. Clearly, this is complementary to the theory-based approach discussed by Westhorp and offers the possibility of interaction between simulation and theory.

In the schematic description of the CoSMoS approach in Fig. 1, the approach to issues with many interacting facets used here is illustrated. There are two cycles to the iteration process. The inner cycle involves comparison of the Results-model generated by the simulation(s) with the Domain-model. This comparison verifies that the simulation produces the key features of the Domain that were captured in the Domain-model. Further iterations of the modelling process are carried out based on this comparison. The outer cycle of the iterative process, validation, allows the modeller to add additional aspects of the system so that progress towards a more complete description of the system is approached in a systematic manner. For example, a more complete description of public health interventions to address health equity require that the general empowerment actions be combined with specific activities to address social determinants of health. In the modelling process described below, the specific actions are excluded and, further, the modelling process is restricted to those aspects that are related to social networking. Subsequent validation stages will consider the degree to which the resulting simulations are capable of describing the system. Extending the boundary of the Domain and hence the content of the Domain-model will allow a more detailed description to be generated.

In the remainder of this Section we describe the Research Context of the modelling process described here (Subsect. 2.2) the construction of the models that underpin construction of the simulation environment (Subsects. 2.3, 2.4, 2.5 & 2.6) and preliminary results of the model (Subsect. 2.7).

2.2 Research Context

The overall research context is that described in the opening sentences of Sect. 1. That is, any intervention intended to improve health equity by addressing the associated social determinants of health occurs in a system that has a unique context. This context consists of the combination of the interactions with the system's external environment and the internal structure of the system. This internal structure consists of the system agents and the relations between them, and reflects the history of the specific system.

Because of the context dependence and because the planner(s) can not know completely the context of the system, i.e. can not completely know the internal state of the system and its subsystems, the outcome can not be predicted with certainty. Rather the planned intervention has an associated spectrum of possible outcomes. Which of these possible outcomes will occur will not be fully determined until the intervention is implemented. Consequently, it is not ntended that the end product of the modelling process be a descriptive tool that predicts a specific outcome given a particular input. Rather, the modelling process aims to construct an exploratory tool capable of allowing planners to investigate possible outcomes and the factors that affect the probabilities of those outcomes. This focus on the probability distribution of a spectrum of possible outcomes reflects the uncertainty of outcome until implementation that exists in the Domain, and the effect the evaluation process has on outcome as described in the next paragraph. This focus on probability of outcomes is also intended to provide an approach which is functional despite the difficulty of defining causality in complex adaptive systems. That is, in systems where the presence of an interaction or set of interactions has an effect on the probability of an outcome but is not causal in a reductionist sense [24]. For further discussion of the potential of a social simulation approach to explore the implications of complex social dynamics see, e.g., Ref [22].

The need for exploratory tools that can investigate potential outcomes of planning choices and associated uncertainties is further underlined by recent developments in the way that evaluations of interventions are conceptualized. Traditional evaluation strategies can be thought of as processes that are parallel to the intervention process but, ideally, do not affect it. The output from such an evaluation process informs external organizations (such as funders or government departments) about progress towards implementing the intervention, completing the intervention, and achieving the desired outcomes. Considerable effort has been directed at identifying effective evaluation strategies for interventions that address inter-related issues such as poverty, poor housing, low education levels, high-risk behaviours etc., either as issues in their own right or in their role as social determinants of health. A key finding from this extensive work is the need to formulate the evaluation strategy as an inherent part of the intervention process [9, 12]. Changing the view of evaluation to incorporate its role as an integral part of the intervention does not remove the traditional role of data collection and analysis to inform external agencies. Rather it adds the responsibility to inform the relevant external agencies, the interveners and the system members during the course of the intervention. This generates considerations of modifications to the intervention that reflect new understandings about the system as they appear. From this viewpoint, the evaluation process facilitates the negotiation between individuals in the system and informs both a reformulation of individual strategies and the strategic framework (often referred to as the Theory of Change) of the intervention. That is, the intervention should itself be considered an adaptive process (Fig. 2) in which evaluation interacts with the intervention planning and implementation. Exploratory tools using in-silico simulation could use information from evaluation to assess potential outcomes of changes made in response to that evaluation and could make the assessment sufficiently quickly to be used as part of the in-progress planning for the intervention.

Consequently there is a need for simulation-based tools that can use data and analysis prior to the intervention, and during the intervention to identify potential outcomes and how changes in strategy and environment affect the probabilities of these outcomes. Moreover, such tools must be able to take into account the specific ontology of a system when making these predictions. We argue here that the building of in-silico environments that allow the exploration of multiple scenarios (including variations of strategies) will provide such tools to facilitate both pre-implementation and in-progress implementation planning.

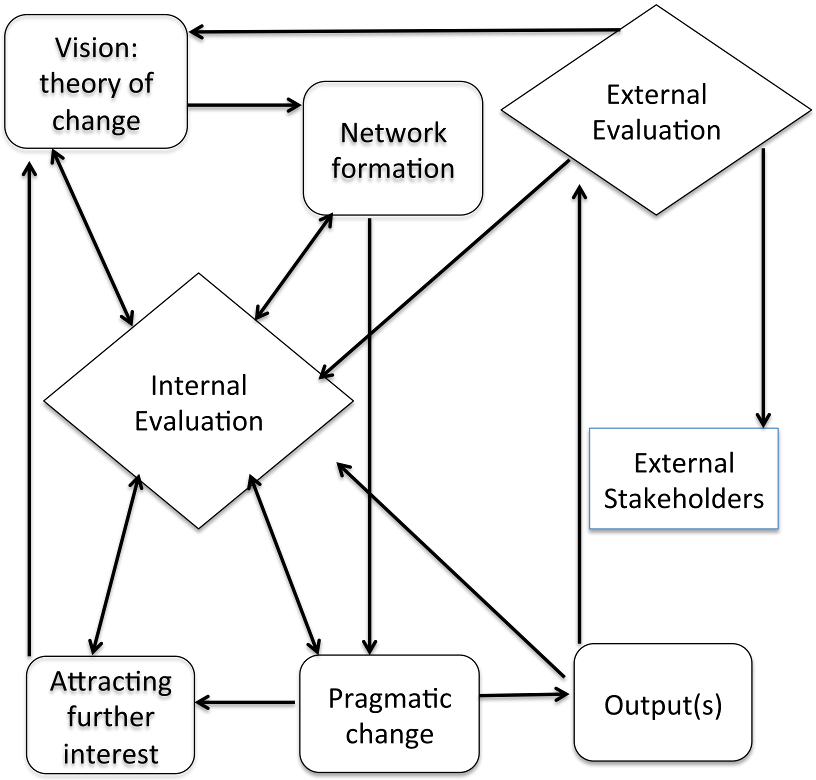


Fig. 2. A schematic representation of the interaction between evaluation process and cycle of change components. See Ref. [5, Section 2.4] for a more detailed illustration of the basic cycle of change structure.

2.3 Domain Definition - a Community Empowerment Approach to Public Health

The domain of the model is defined by specifying those aspects of the real-world that the software is to simulate. For the purpose of developing an in-silico simulation tool, we define the domain as community empowerment interventions addressing the social determinants of health. This approach has been advocated by public health researchers (see, for example, Ref. [18] and references therein). Community empowerment can be seen as a process that runs in parallel with and interacts with traditional top-down public health activities that address specific issues such as vaccination programs and anti-obesity programs. Tensions arise between this empowerment stream in public health programme planning and the more conventional stream of public health programme planning. However, an integrated strategy allows these tensions to be made explicit and decisions made on how to deal with them at each stage of the planning and implementation process [17]. In a similar way, community empowerment is complementary to addressing specific underlying social issues such as implementing a living wage policy [19], tackling health related worklessness [4], and building age-friendly communities [16] (see Ref. [7] for further examples).

A clear definition of the Domain is required in order to place a boundary around the modelling process. Expansion of the Domain can then be considered as part of the iterative process e.g. at a later stage the inclusion of specific types of action can be included. To clearly state the boundary of the domain we specify the following conditions.

- Programmes dealing with specific diseases or prevention measures, such as an inoculation programme or obesity reduction programme, are excluded from the domain. We acknowledge that such specific actions may be part of a PH intervention in which the planning tool is used. Therefore, the planning tool or, at least, its applicability should be sufficiently flexible that it can incorporate very specific actions. However, for definiteness, we exclude such actions here.
- Included in the domain are interventions that are intended to address social inequities through community empowerment even when these interventions are not specifically addressing the relation that these social inequities have with health inequities. For example, the domain will be taken to include interventions to address poverty, low education attainment, poor quality housing and/or other interrelated factors even when the primary concern of the intervention is not health in the narrower interpretation of this term. Specific actions or policy changes are not included at this stage.

The choice of which experts or which expert knowledge are consulted also determines the domain of the model . The eventual expansion of the domain to include a greater range of expert knowledge can be considered part of an iterative process. Public health expertise is distributed among public health researchers, public health planners, public health decision makers and public health evaluation experts. Further public health knowledge is held by front-line public health workers and public health clients. Front-line public health worker knowledge and public health client knowledge contains elements that are specific to the particular (local) community within which the intervention occurs. Inclusion of this local specific knowledge in the domain will result in models specific to this community. Therefore we need to construct generic models such that local expert knowledge can be incorporated into the simulation. For the purpose of an initial model construction we draw on published reports of public health interventions and social change in which community empowerment has been a major consideration. Iteration to expand the domain would include consultations with knowledge experts in both policy and public health.

We restrict the domain description to the following observations that are consistent with the literature reviewed.

- Interventions require the engagement of multiple stakeholders and layered structures (e.g. national or regional stakeholders, intermediaries responsible for the "on-the-ground"intervention, and community stakeholders) [see, e.g., Ref. [11]]
- Social network structures are built and maintained through individual-to-individual contacts (even when the formal description of the connection is in terms of organization-to-organization agreements). Social distance is a determining factor in the formation of social connections. Social influence between individuals is dependent on a number of factors [8]. These factors include physical distance and psychological distance. Psychological distance incorporates perceived differences in social factors such as status, authority, social standing and power. Here we take the term "social distance"to mean the combined effect of physical distance and social/psychological factors that create social separation between individuals.
- Community empowerment interventions seek to increase community capacity to access and manage resources, and to influence decisions that affect the community [18].

2.4 Model Framework: the Domain Model

The domain-model is constructed as a representation of the domain that contains key properties of the domain including the underlying structure and dynamics, and the list of observables relevant to the domain.

We represent the multiple stakeholder, layered structure of the system as a three layer process with in-layer and between-layer communications carried by social network structures. Each layer consists of individuals that may form network connections with others in their own layer and with individuals in the other layers (Fig. 3). The three layers of the system are listed in Table 1.

Network dynamics are determined by the following

- Connections between individuals are formed, build in strength and are maintained through social meetings.

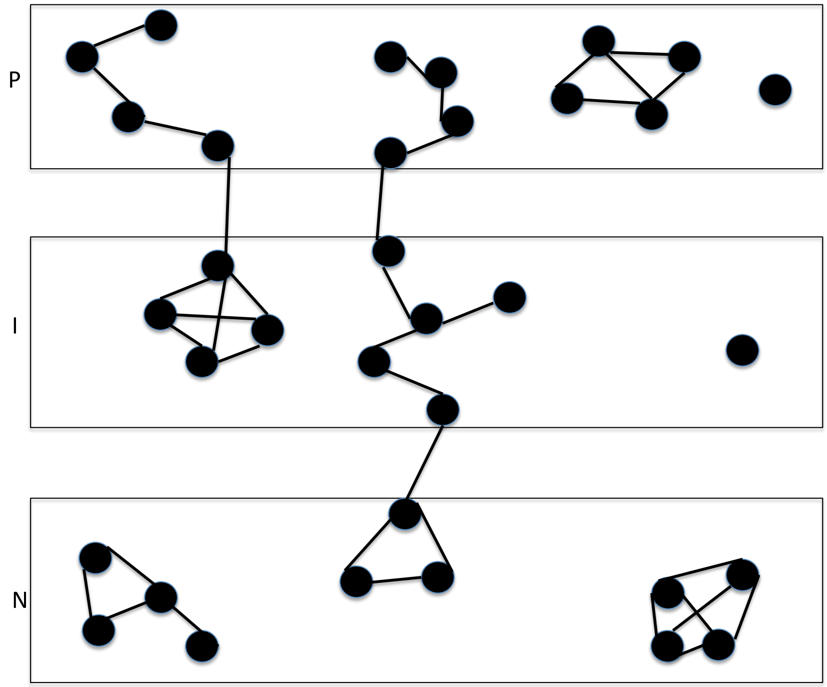


Fig. 3. Schematic representation of the three layers of the domain model: Political/Policy (P), Intervener/Intermediate (I), and Neighbourhood (N), and network structure

– Social distance determines which individuals will form (direct) connections
– Connections that are not maintained fade and will eventually cease.

To complete the domain-model we specify the set of observables that are to be generated in the simulations. Laverack [6] lists a set of nine empowerment outcomes ("domains of empowerment"). From these nine outcomes, we select four observables that can be related to network properties (Table 2). The translation of these observable properties to measurable quantities specified in the platform model may be considered as analogous to the translation of outcomes to measurable indicators that occurs in constructing an evaluation scheme. As with indicators used in the domain any observable specified in the platform model must be measurable.

Table 1. Layers of the 3-layer Domain-model

Layer	Description
Political/Policy (P)	Agencies which control funding, resources and policy (including, e.g., non-government funders) at a national or regional level
Intervener/Intermediary (I)	Agencies and individuals that act to implement the changes directed at community empowerment
Neighbourhood (N)	Individuals and agencies that act locally within the communities that the empowerment program (and related public health initiatives) act in.

Table 2. Translation of the Domain-model community outcomes to platform model outputs. Domain outcomes are selected from the nine "domains of community"identified by Laverack [18]

Domain Outcomes	Translation	Platform-model Outcomes
Develops local leadership	Neighbourhood leaders must have both intra and inter layer connections	Number of N layer individuals with both intra-layer and interlayer connections. Individuals with both types of connection are identified as (neighbourhood) leaders
Strengthens links to other organizations and people	Leaders must have and maintain links to other individuals	Number of other individuals connected to each leader (in all layers)
Building empowering organizational structures	Leaders must form and maintain connections with the intervener and political levels	Number of individuals in the P and I layers that leaders are connected to
Increases control over program management	Leaders must form and maintain connections to the political layer	Number of individuals in the P layer connected to leaders

2.5 Platform Model

The platform model provides a description of the system that can be implemented as software. Here, we implement the system as a network of agents (nodes in the network) distributed in a "social space". (There is an extensive literature on the use of network models for the modelling of social structures. We refer the reader to Ref. [25] for a summary of some key articles).

Each agent is assigned to one of the three layers of the domain model (P, I & N). The location of each agent in social space is represented by allocating the agent a (x,y) coordinate in a (bounded) plane representing the corresponding layer of the domain-model. The relative position of the three planes in this social space is determined by assigning a z-coordinate to each of the planes. Adjacency (direct connection) is initially determined by social distance. If the distance between any two agents is less than a specified value a direct connection between them is assigned as a nonzero adjacency weight ($0 \leq w_{ij} \leq 1$).

Evolution of the system occurs through (Markov) dynamics consisting of three actions in each time step:

- Random selection of a subset of agents who invite the agents to whom they are directly connected to a meeting.
- An increase in the adjacency-weight by a set amount, up to a maximum weight of 1, for all agent-pairs with an inter-agent distance less than a specified value if both agents in the pair attend a meeting. This allows agent-pairs that were previously connected but not adjacent to acquire a non-zero adjacency weight. The maximum social distance specified for this step may be greater than (or less than) the maximum social distance over which direct connections are formed in the initial network construction.
- A decrease in all non-zero adjacency-weights by a specified amount.

This dynamical process is based on an algorithm for social network dynamics recently described by zu Erbach-Schoenberg et al. [25]. An advantage of the algorithm introduced by zu Erbach-Schoenberg et al. stems from the use of adjacency weights (network edge weights). This enables a gradual change in the strength of connections between pairs of agents as connections are formed and maintained through meetings (the first and second step above), or decline due to lack of maintenance (the third step above). As noted by zu Erbach-Schoenberg et al., basing initial and evolving adjacency on social distance combined with agents with static positions has a limiting effect on the building of network [25]. That is, the dynamical process of forming direct connections through the agency of a "mutual friend"(i.e. the second item in the list of actions above) means that agents can not become adjacent to or connected to agents that they were not connected to in the initial network in the implementation of the algorithm used here. zu Erbach-Schoenberg et al. have implemented the algorithm as a model of spatially constrained systems and performed a comprehensive study of the network properties in a two dimensional realization [25]. The implementation of the algorithm used here differs in some details from the implementation of zu Erbach-Schoenberg et al. One such difference is that this implementation does not use the “affinity”parameter, introduced by zu Erbach-Schoenberg et al, which is the probability that two individuals make a connection at a meeting. Rather the distancing that may be caused by social/psychological factors is assumed to be included in the “social-distance”as described above. This has the advantage of reducing the number of parameters in the model however it raises questions about calibration that we return to in Sect. 3.

The observables of the platform model are identified by interpreting the domain outcomes in terms of network connectedness (Table 2). The validation process tests the degree to which the domain outcomes can reasonably be represented by network statistics. We comment on this further in Sect. 3. In what follows, ”connectedness” is used to indicate the number of other agents that a given agent is connected to; i.e., the number of other agents in the component to which the specified agent belongs. As usual ”degree” indicates the number of other agents that a given agent is directly connected to (adjacent to).

2.6 Software Implementation - The Simulation-platform

The in-silico experimental environment used to experimentally explore the system behaviour is generated by constructing the Simulation platform, a software representation of the Domain-model. For the purpose of the illustrative development of a network model here, a Fortran 90 Simulation-platform was constructed. The outputs from the simulations correspond to the simple connectedness indicators outlined in Table 2.

The Simulation-platform is constructed so that the number of agents in each layer can be specified for a given simulation. Values of the model parameters, such as the social distance used to determine the initial network and the distance over which direct connections can be established/strengthened during meetings, are also input for a given simulation of the system. A single simulation run consists of the generation and evolution of a specified number of networks with fixed values for the model parameters.

2.7 Simulation Output - the Results-model

The results model consists of the description of the system obtained from the simulation outputs. We present here some preliminary results that illustrate the outputs of the current model.

Connectedness statistics, corresponding to the four Domain-model observables, were generated by implementing the platform as code and assigning the program parameter values. Parameter values were chosen for convenience in assessing the properties of the model. Consequently, the description generated is illustrative rather than corresponding to any specific system of the domain. Each layer plane has a square geometry with a linear dimension of 200. The maximum social distance for the initial determination of adjacency was assigned a value of 20. The maximum social distance for adjacency via the friend-of-a-friend mechanism (i.e. for agents that meet due to a mutual friends invitation) was assigned a value of 40. The z coordinates of the P, I and N planes were assigned values of 14,0,-14 respectively. I.e. the parameters were deliberately chosen so that agents in the N and P layers are not adjacent in the initial network but adjacency between these agents may evolve through the social network dynamics in the platform-model. This choice corresponds to the understanding that community empowerment can be thought of in terms of connecting individuals to the political and policy processes. The number of agents in the N and P layers were set to 100 and 5 respectively. The number of agents in the I layer was varied from 0% to 24% of the number of agents in layer N.

Statistics were collected for the initial network and final network for each value of the number of layer I agents considered. For completeness, in Fig. 4 we show the behaviour of the average of both the initial and final statistics corresponding to the connectedness and degree of the neighbourhood leaders, as the number of agents in layer I is varied. (Averages are over 100 networks). Each data point was generated by a single simulation run. The averages in Fig. 4 illustrate certain general features of the model. In particular, comparison of the initial and final (total) connectedness and (total) degree of agents identified as community leaders shows that the total connectedness decreases but the degree increases with time. I.e. the networks of leaders decrease with time but those that are sustainable become much more complete (more direct connections). While the averages are useful for illustrating general behavior they obscure one of the key features of the results, that is that the survival of a given leader's network depends on the initial structure of the network, i.e. the final structure of the network may vary widely between networks with a large highly connected component which contains all or most of

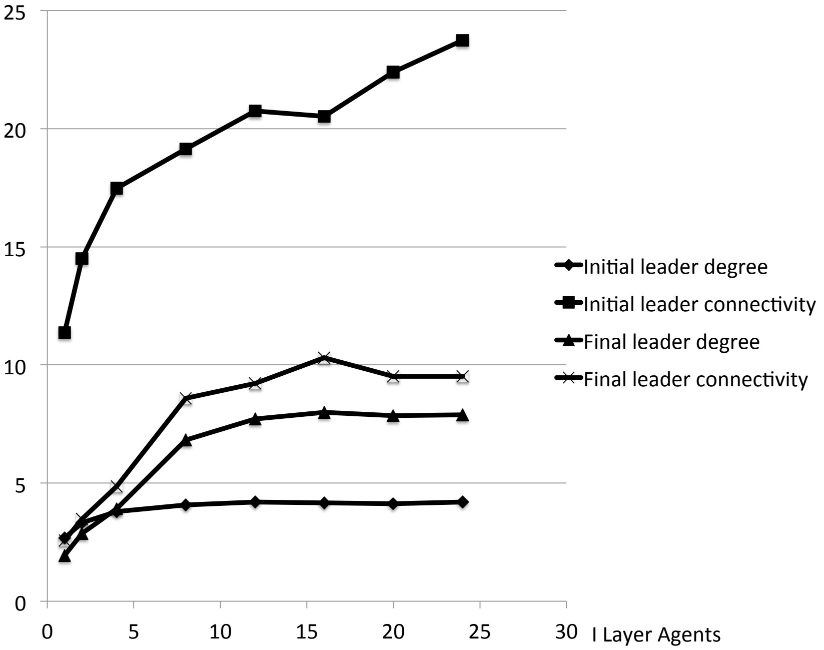


Fig. 4. Average connectedness (component size) and degree (number of direct connections) of community leaders in the network model vs. number of agents in the intervener (I) layer as a percentage of the number of agents in the neighbourhood layer (N).

the leaders, to a network of essentially isolated individuals depending on the structure of the initial network.

Although the general trend in the network statistic with the number of agents in layer I can be discerned from Fig. 4, there is clearly some variation of the data points around a smooth line that might be drawn to represent the general trend. This is an indication of the wide variation in characteristics of initial networks generated with the model parameters specified above, and the corresponding variation in the final networks. The frequency distribution of the network average of initial and final leader connectedness across the 100 networks corresponding to the MI=24 data point in Fig. 4, is shown in Fig. 5. As can be seen in Fig. 5, the distribution of network averages of connectedness for leaders for the initial networks has a long tail with small but significant population in the tail of the distribution. The evolution of the networks results in a distribution of final network average component sizes for leaders that is

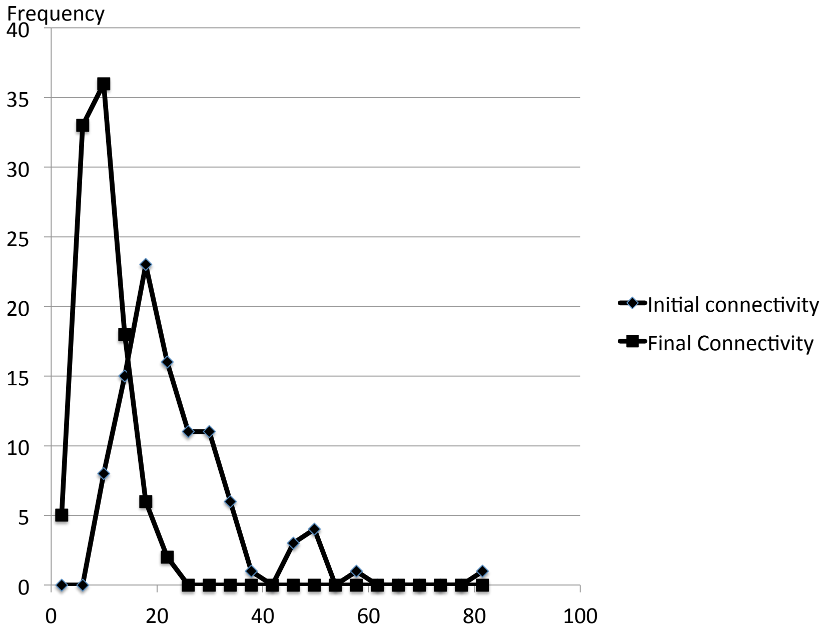


Fig. 5. Frequency distribution of the initial and final neighbourhood leader network average connectedness across 100 networks with 24 intervener (Layer I) agents.

narrower and shifted to smaller values on average. The wide distribution of network averages is related to the choice of parameters which, in this case, led to a wide variation in the network structures. This variation in the initial network structure can be reduced by, for example, increasing the density of agents or increasing the social distance over which direct connections form, in such a way that the number of components is small and almost all agents belong to a large component.

With a long tailed broad distribution of leader connectedness, such as that shown in Fig. 5, an assessment of error bounds for data points shown in Fig. 4 requires some care. To ensure a meaningful estimate of error bounds the mean and statistical deviation of the simulation run average for each statistic (e.g. leader connectedness) was calculated over ten separate simulation runs. Each of the simulation runs consisted of 100 networks. Sample results for networks with 24 agents in the I layer are shown in Table 3. The error bars are ± two standard deviations.

For the parameter set chosen here, the average number of leaders decreases as the system evolves. However, individual networks may evolve

Table 3. Mean and error bars for the simulation average of the initial and final leader connectedness in systems with 24 agents in Layer I. Estimates of the mean and error bars were calculated using ten simulations. Each simulation consisted of the generation and evolution of 100 networks

	Initial	Final
Leaders	28.7 ± 1.2	17.2 ± 2.2
Degree of Leaders	4.2 ± 0.1	8.0 ± 0.6
Connectedness of Leaders	25.3 ± 2.1	9.6 ± 0.9

to have a smaller number of leaders or a greater number of leaders than were initially present in the network. Moreover, the set of leaders in the final network are not necessarily a subset of the leaders in the initial network (though the two sets must have at least one agent in common). That is, due to the effect of the competing actions of building social connections through meetings and connections declining through lack of maintenance, initial leaders may be "lost" (cease to be leaders) but new leaders may emerge due to their connections with existing leaders.

The average behaviour of the remaining platform model observables (connections to agents in the P and I layers, and connections to agents in the P layer only) again indicate some general properties of the model. The behaviour of the connections to outside partners is consistent with that observed for all connections i.e. for the parameter set chosen, the average number of agents in layers P and I connected to a leader (the leader's number of external partners) decreases as the network evolves but the number of direct connections a leader has to agents in layers P and I increases. Connectedness to the P layer by neighbourhood leaders remains very low on average even at the highest value of the number of I layer agents considered.

3 Discussion

The process of modelling community empowerment outlined in Sect. 2 results in a 3 layer (process) model with communication between the layers carried by social connections. The platform model consists of an abstract network of agents with connectedness that evolves through a dynamical process with two competing elements. The first of these elements builds direct connections between agents (network nodes) through a process that simulates the initiation and maintenance of social connections through individual-to-individual contacts (meetings). The second element diminishes direct connections between agents and simulates the decay of social connections that occurs if individuals do not maintain their social relationship through contacts (meetings) [25].

Observables measured in the simulation are expressed in terms of connectedness in the network. These observables were identified by considering the community empowerment outputs ("domains of community empowerment") identified by Laverack [18].

The parameters used to generate preliminary results from the simulations were chosen for convenience and to reflect reasonable general assumptions about the dynamical process. For example, the maximum social distance over which direct connections can be established following introduction by a mutual friend was specified to be twice that of the maximum social distance over which initial direct connections can be formed. This reflects the property of social systems that introductions by a mutual friend allow an individual to make direct contacts with individuals that they might not otherwise connect with directly. Parameterizing the model in this way allows us to determine certain general properties of the model and to ask how those properties compare with data from actual community empowerment interventions. These considerations identify two interconnected problems that confront attempts to model social systems to a level which will provide context specific simulations; i.e.

- Is it possible to map concepts such as social distance onto simple model variables? If so, what is the appropriate form of the variable? Here we have used a three dimensional vector to represent social distance and, in particular, to separate the different layers in the system. It is, of course, possible to consider using higher dimensional vectors to represent the multiple factors that contribute to social distance [8]. However, it remains to be tested if such a higher dimensional representation would improve the model in a significant way.
- If appropriate model variables can be identified, how is the model calibrated? (See also Ref. [25] for comments on the calibration of models involving spatial separation of agents).

Returning to the general properties of the model indicated by the preliminary results presented here, we identified that the surviving network components connected to agents identified as community leaders tend to shrink but to evolve towards a highly connected group. While this is apparent from the averaged data in Fig. 4, the presentation of average data hides certain information that the simulations reveal when a more detailed analysis is conducted. In particular there is a dichotomy among the community leaders. Initial network components associated with the community leaders tend to evolve to one of two types of structure; i.e. after sufficient time steps, either the component the leader is connected to is highly connected and consequently quite stable, or the leader is isolated with no (or very few) remaining connections. Clearly a choice of model parameters that favour one of these contrasting behaviours over the other can be made. For example increasing the number of meetings per time step while holding the decrease due to lack of meeting in a given time step constant, will tend to favour the formation of stable networks. In this sense the model is consistent with the observation that the convener role of agencies in the intermediate group in the early stage of an intervention, facilitates the building of momentum and eventual sustainability of the initiative [11].

For a given choice of parameters, which of the two groups of the dichotomy a given leader evolves to is determined by the initial structure of that leader?s network component as well as the specific set of random numbers that determine which leaders issue meeting invitations. This is, of course, consistent with our expectation of context dependent behaviour in complex systems. However, it implies that if we are to develop simulation tools that can identify the spectrum of possible outcomes, knowledge of the initial network structure is particularly important. Network mapping is a commonly used way of visualizing and characterizing social networks. The point here is that network mapping information as input to the simulation appears to be essential for accurate forecasting of the spectrum of possible outcomes for an intervention in a specific system i.e. local knowledge such as may be held by frontline local public health workers and neighbourhood residents is critical to provide the necessary context information.

An advantage of the CoSMoS approach is that it makes explicit the modelling decisions that are made at each of the stages of the modelling procedure. Modelling decisions made in the application of the process described above included modelling individuals as agents that are fixed at randomly allocated locations in a social space. Other than their locations, which codify their association with a particular layer, all agents in the platform-model are equivalent in that they each use the same mechanism for creating, building and maintaining social connections and each does so with the same probability of calling a meeting of friends. Consideration of the extension of the model by having agents with different characteristics raises the question of the network building dynamics. In particular, it appears necessary to consider if some agents may use strategies that are both intentional and that are less limited by considerations of social distance.

Acknowledgements The author acknowledges with thanks, support for this research provided by the Department of Computer Science, University of York, the York Centre for Complex System Analysis (YCCSA), and St. Francis Xavier University. It is also a pleasure to thank Susan Stepney (YCCSA) for discussions related to the application of the CoSMos approach and insightful comments on the approach taken here. Thanks also to Leo Caves, John Forrester, Dan Franks, Richard Law, Fiona Polack (YCCSA), Richard Taylor (Stockholm Environment Institute, Oxford Centre), Kate Pickett and Stephanie Prady (Health Sciences, University of York), and David Hunter and Emily Henderson (Centre or Public Policy and Health, Durham University) for discussions and correspondence.

References

[1] A. Afshar Dodson, S. Stepney, E. Uprichard, and L. Caves, "Using the CoSMoS approach to study schelling's bounded neighbourhood model," in *Proceedings of the 2014 Workshop on Complex Systems Modelling and Simulation*. Luniver Press, 2014.

[2] P. S. Andrews and S. Stepney, "Using CosMoS to reverse engineer a domain model for Aevol," in *Proceedings of the 2014 Workshop on Complex Systems Modelling and Simulation*. Luniver Press, 2014.

[3] P. S. Andrews, S. Stepney, T. Hoverd, F. A. C. Polack, A. T. Sampson, and J. Timmis, "The CoSMoS process, models and metamodels," in *Proceedings of the 2011 Workshop on Complex Systems Modelling and Simulation*. Luniver Press, 2011, pp. 1–13.

[4] C. Bambra, "Tackling health-related worklessness: a 'health first' approach." The British Academy for the Humanities and Social Sciences, 2014.

[5] H. Bevan, P. Plsek, and L. Winstanley, *Leading Large Scale Change: A Practical Guide. What the NHS Academy Learnt and How You can apply these Principles within Your own Health and Healthcare Setting.* NHS Institute for Innovation and Improvement, 2013.

[6] J. Bown, P. S. Andrews, Y. Denmi, A. Goltsov, M. Idowu, F. A. C. Polack, A. T. Sampson, M. Shovman, and S. Stepney, "Engineering simulations for cancer systems biology," *Currrent Drug Targets*, vol. 13, no. 12, pp. 1560–1574, 2012.

[7] British Academy for the Humanities and Social Sciences, *If You Could Do One Thing: Nine Local Actions to Reduce Health Inequalities*. The British Academy for the Humanities and Social Sciences, 2014.

[8] O. Burak and M. R. Bashur, "Followership, leadership and social influence," *The Leadership Quarterly*, vol. 24, pp. 919–934, 2013.

[9] M. Cabaj, "Evaluating collective impact: Five simple rules," *The Philanthropist*, vol. 26, no. 1, pp. 109–124, 2014.

[10] B. E. Cohen, A. Schultz, E. McGibbon, M. VanderPlaat, R. Bassett, K. GermAnn, H. Beanlands, and L. A. Fuga, "A conceptual framework of organizational capacity for public health equity action (OC-PHEA)," *Can. J. Public Health*, vol. 104, no. 3, pp. e262–e266, 2013.

[11] J. Gamble, *Inspired Learning: An Evaluation of Vibrant Communities National Supports.* Tamarack, An Institute for Community Engagement, Waterloo, Ontario, 2012.

[12] J. A. A. Gamble, *A Developmental Evaluation Primer.* The J.W. McConnell Foundation, 2008.

[13] R. B. Greaves, F. A. C. Polack, and J. Forrester, "CoSMoS in the context of social-ecological systems research," in *Proceedings of the 2012 Workshop on Complex Systems Modelling and Simulation.* Luniver Press, 2012.

[14] I. Greener, B. E. Harrington, D. J. Hunter, R. Mannion, and M. Powell, *Reforming Healthcare: What's the evidence?* Policy Press, Bristol, 2014.

[15] D. J. Hunter, *The Health Debate.* The Policy Press, Bristol, 2008.

[16] H. Kendig and C. Phillipson, "Building age-friendly communities: New approaches to challenging health and social inequalities." The British Academy for the Humanities and Social Sciences, 2014.

[17] G. Laverack, *Public Health: Power, Empowerment and Professional Practice.* Palgrave MacMillan, Hampshire and New York, 2005.

[18] ——, *Health Promotion Practice: Building Empowered Communities.* Open University Press, McGraw Hill, New York, 2007.

[19] K. Pickett, "Addressing health inequalities through greater social equality at a local level: Implement a living wage policy." The British Academy for the Humanities and Social Sciences, 2014.

[20] D. Raphael, Ed., *Social Determinants of Health: Canadian Perspectives*, 2nd ed. Canadian Scholars' Press, Toronto, 2009.

[21] D. Raphael, *Poverty in Canada: Implications for Health and Quality of Life*, 2nd ed. Canadian Scholars' Press, Toronto, 2011.

[22] F. Squazzoni, W. Jager, and B. Edmonds, "Social simulation in the social sciences: A brief overview," *Social Science Computer Review.*

[23] S. Stepney and P. S. Andrews, "CoSMoS special issue editorial," *Natural Computing*, vol. 14, no. 1, pp. 1–6, 2015.

[24] G. Westhorp, "Using complexity-consistent theory for evaluating complex systems," *Evaluation*, vol. 18, no. 4, pp. 405–420, 2012.

[25] E. zu Erbach-Schoenberg, S. Bullock, and S. Brailsford, "A model of spatially constrained social network dynamics," *Social Science Computer Review*, vol. 32, pp. 373–392, 2014.

Genome-wide mouse embryonic stem cell regulatory network self-organisation: a big data CoSMoS computational modelling approach

Richard B. Greaves[1], Sabine Dietmann[2], Austin Smith[2], Susan Stepney[1], and Julianne D. Halley[1]

[1] York Centre for Complex Systems Analysis, University of York, UK
[2] Wellcome Trust-Medical Research Council Cambridge Stem Cell Institute, University of Cambridge, UK

Abstract. The principal barrier to gaining understanding of embryonic stem (ES) cell regulatory networks is their complexity. Reductionist approaches overlook much of the complexity inherent in these networks and treat the ES cell regulatory system as more or less equivalent to the sum of its component parts, studying them in relative isolation. However, as we learn more about regulatory components it becomes increasingly difficult to integrate complex layers of knowledge and to develop more refined understanding. We seek better control of the complexity inherent in non-equilibrium ES cell regulatory networks undergoing lineage specification by developing computer simulations of self-organisation using the CoSMoS approach. Simulation, together with the hypothesis that lineage computation occurs at the edge of chaos, should allow us to investigate the driving of gradual accumulation of network complexity 'from the bottom up'. Here, we present the first step in this design process: use of the CoSMoS approach to develop a highly abstracted model and simulation of regulatory network activity driven by just single pluripotent transcription factors (TF), but at genome-wide scales. We investigate three TFs in isolation: Oct4, Nanog and Sox2, central elements of the core pluripotent network of mouse embryonic stem cells. This provides a suitable basis for future modelling of multiple interacting TFs.

1 Introduction

Mathematical or computational frameworks and tools are indispensable in the study of cell regulatory networks [12, 51] because functions, traits and pathologies are rarely caused by single genes [12, 26, 49]. However, the principal challenge that prevents comprehensive understanding (and simulation) of regulatory networks is their complexity [36]. Indeed, in the era of systems biology, the icon for molecular biology is the 'hairball' graph, which illustrates how everything seems to interact with almost everything else [20, 32]. High-throughput technologies generate such large volumes of data that there is concern about how to grasp the big picture [14, 17, 27] and most data sets are not being used to their full potential.

Here we present the first iteration of a novel computational framework to interrogate the complexity of stem cell regulatory networks. We employ a previously described theoretical framework based on the notion that the backbone of stem cell fate computation is provided by the critical-like self-organisation of transcription factor (TF) regulatory networks [23–25].

We apply the modelling framework CoSMoS [3, 43, 45], which is specifically designed to capture the emergent properties of complex systems, and to guide the engineering of trustworthy computer simulations, i.e., those that are scientifically valid, useful and credible to third parties.

The models and results in this paper report on the first iteration of the CoSMoS design cycle. Here, we design and calibrate simulations of single TFs in isolation. This single TF version of the full model is not biologically realistic; its purpose is to serve as a building block of complexity that will be iterated in our next work.

The structure of the paper follows the patterns defined in the CoSMoS approach outlined in §2. This progresses through the definition of scope and the model of the scientific domain in §3, then the development of the simulation software in §4, and use of the simulation to run experiments and explore system behaviour in §5. We conclude with some reflections on the process in §6, and discussion of further work in §7.

2 The CoSMoS approach

The CoSMoS approach [3, 43, 45] enables the construction and exploration of computer simulations for the purposes of scientific research. It describes a series of models and other components that need to be specified, designed, and implemented in order to build and use a fit-for-purpose simulator. The approach is guided by considering the simulator to be a form of *scientific instrument* [6] that needs to be carefully designed, built, calibrated and used in a manner appropriate to the specific research questions.

The CoSMoS approach is encapsulated as a *pattern language* [1]. The CoSMoS patterns provide guidance on what to do at the various stages of a CoSMoS simulation project [42, 45]. We structure this paper explicitly in terms of these patterns.

To guide the reader through the pattern structure, we reproduce in boxed text a brief overview of the pattern: the pattern name and *intent*, a short phrase describing what should be done; and, where applicable, any *components* (including sub-patterns) that can be used to decompose the intent. We use section subheadings to capture the specific pattern names (named with initial capitals, such as *Research Context*) and other components (named in lower case, such as *success criteria*) and their position in the overall pattern structure.

We start at the top level of a simulation project, which is formed of three phases per iteration of the project.

CoSMoS pattern: *CoSMoS Simulation Project:* Develop a basic fit-for-purpose simulation of the complex scientific domain of interest.

The components of a *CoSMoS Simulation Project* are:

– carry out a *Discovery Phase*
– carry out a *Development Phase*
– carry out an *Explorations Phase*
– iterate as required

In this paper we report on the first iteration of our simulation, comprising a simulation of a single TF branching process. This provides the basis for the next iteration, which will add multiple interacting TFs. The next three sections document the results of carrying out this first iteration of each of these three phase patterns, structured in terms of their sub-patterns.

3 Discovery phase

CoSMoS pattern: *Discovery:* Decide what scientific instrument to build. Establish the scientific basis of the project: identify the domain of interest, model the domain, and shed light on scientific questions.

The components of the *Discovery* phase are:

– identify the *Research Context*
– define the *Domain*
– do *Domain Modelling*

- define the *Expected Behaviours*
- Argue Appropriate Instrument Designed (omitted here)

3.1 Discovery > Research Context

CoSMoS pattern: *Research Context:* Identify the overall scientific context and scope of the simulation-based research being conducted.

The components needed to identify the *Research Context* are:

- provide a thumbnail *overview* of the research context
- document the *research goals* and project scope
- agree the *Simulation Purpose*, including criticality and impact
- identify the *Team* members, including the Domain Scientist, the Domain Modeller and the Simulation Engineer, their roles, and experience
- document *Assumptions* relevant to the research context
- note the available *resources*, timescales, and other constraints
- determine *success criteria*
- decide whether to proceed, or walk away

Discovery > Research Context > overview

The context of this research is the investigation of a conceptual approach: self-organisation at the edge of chaos. We have argued that if the activity of single transcription factors can be described as critical-like branching processes, their interplay should define a critical-like genome-wide interference pattern that captures in some way the nature of the entire pluripotency transcription factor regulatory network [24].

Here we build a simulation based on the representation of TFs as *branching processes*. The mathematical concept of a branching process (BP) is as follows. Consider a population of individuals. At time t each individual i produces a next generation of m_i offspring individuals, with the value of m_i drawn from some probability distribution. Let the average number of offspring produced be μ. If $\mu > 1$, then the process is supercritical and the number of individuals grows without bound. If $\mu = 1$ then the system is critical and can either give rise to more individuals in the next step or lead to dissipation of the process. If $\mu < 1$ then the process goes to extinction.

Our model of TF BPs builds on this idea, and also allows the TFs to *interact* in such a way as to cause the regulatory network to self-organise at the edge of chaos. We capture the activity of single TFs as BPs in order to predict the interplay of multiple TFs and the emergent nature of the entire TF regulatory network, hypothesised to operate in a critical-like state [24].

For a TF to be stably expressed, its BP must be supercritical [24]. Therefore, by modelling the activity of TFs known to be expressed in mouse embryonic stem cells, we link the perturbation of a TF's cistrome (portion of the genome in which the TF displays some activity) with a dynamic and distributed description of TF activity. This is a prerequisite to being able to simulate the entire TF regulatory network of an ES cell, as argued in [24]. The TFs called Oct4, Sox2 and Nanog are central elements of the core pluripotent network of mouse embryonic stem cells. In the first instance, the current work will allow us to calibrate our simulation for these three TFs in isolation, that is, to characterise how their associated TFBPs propagate in the absence of interference.

Our iterative approach to the development of the full simulation commences with the simplest possible system: the operation of one transcription factor at genome-wide scales. We will later add layers of further complexity, testing and calibrating as we go.

A model of a single pluripotent TF in isolation is far from complete and is not biologically realistic. It is only when multiple TF BPs are simulated in parallel that we can expect to generate the interference patterns predicted to underpin circuitry self-organisation. As greater numbers of pluripotency TFs are included in the model, we anticipate that our simulations will become increasingly biologically realistic. In future work we will augment the complexity of the computational model in a stepwise manner, adding detail and refining assumptions as we progress, and increasingly be able to provide insights not accessible by other means.

Discovery > Research Context > research goals

The overall research goals of this work are:

1. to create a simulation of Branching Process Theory (BPT) as applied to embryonic stem cell differentiation
2. to use this simulation to validate the application of BPT in this context
3. to make the simulation available for more general use

Here we report on the first iteration, of a single TF branching process.

Discovery > Research Context > Simulation Purpose

CoSMoS pattern: *Simulation Purpose:* Agree the purpose for which the simulation is being built and used, within the *Research Context.*

The components of the *Simulation Purpose* are:

- define the role of the simulation
- determine the criticality of the simulation results

Simulation role: The role of the simulation is exploratory: to provide evidence of the usefulness of BPT as a model of decision making in stem cell differentiation. The simulation will be used to investigate which values of the average branching ratio are required to set up a sustainable TF branching process.

Simulation criticality: The simulation work is being used to explore the suitability of of a particular approach, BPT, in the domain. The simulation results are not safety, security, or financially critical: they will not be used directly in the development of any products.

Discovery > Research Context > team

The three main CoSMoS roles are fulfilled by the team members in the following way:

- *Domain Scientist*: Halley, an expert on BPT as applied to stem cell differentiation, backed up by a domain expert in ES cell biology (Smith), and a data collection expert (Dietmann)
- *Domain Modeller*: Greaves, with CoSMoS domain modelling experience, backed up by a further CoSMoS modelling expert (Stepney)
- *Simulation Engineer*: Greaves, with agent based simulation engineering experience

Discovery > Research Context > Assumptions

CoSMoS pattern: *Document Assumptions:* Ensure assumptions are explicit and justified, and their consequences are understood.

The components of *Document Assumptions* are:

- identify that an assumption has been made, and record it
- for each assumption, determine its nature and criticality
- for each assumption, document the reason it has been made
- for each reason, document its justification, or flag it as “unjustified” or “unjustifiable”
- for each assumption, document its connotations and consequences
- for each critical assumption, determine the connotations for the scope and fitness-for-purpose of the simulation
- for each critical assumption, achieve consensus on the appropriateness of the assumption, and reflect this in fitness for purpose arguments

- revisit the simulation scope in light of the assumption, as appropriate

A.1 Cistrome data can be provided by processed ChIP-Seq data
reason It is the data we have
justification This is one standard use for ChIP-Seq data
consequence ChIP-Seq data is variable across measurements, so we will need to check the robustness of our results to this variation

A.2 It is sufficient to consider only the key pluripotency transcription factors: Nanog, Oct4, Sox2
reason As a first step in providing insight, we consider the three TFs widely acknowledged to be central components of the core pluripotent network
justification See for example [13]
consequence We will not be able to determine the effect of further TFs. However, it should be straightforward to incorporate further TF data into the multi-cistrome model.

A.3 We can use mouse data as a suitable proxy for data from human ES cells
reason Suitable mouse data is more readily available; mouse ES cells have an unambiguous ‘ground state’; so mouse data is a good basis for evaluating the TF BP model
justification Although effective manipulation of human ES cells is a long term goal, here we are only assessing the TF BP model
consequence We cannot extrapolate results to the human system

Discovery > Research Context > resources, timescales, other constraints

The project has a one year duration. The Domain Scientist is employed full time, and Simulation Engineer part time.

The work has access to a local computer cluster, for running simulations and gathering performance metrics.

The team members are split between York (Halley, Greaves, Stepney) and Cambridge (Smith, Dietmann)

Discovery > Research Context > success criteria

1. a single-cistrome simulator that exhibits the expected behaviours, and can be used as the basis for multi-cistrome simulator development
2. a multi-cistrome simulator that can justify the use of the TF BP model to analyse stem cell fates

This paper documents the first iteration: the single-cistrome simulator

3.2 Discovery > Domain

CoSMoS pattern: *Domain:* Identify the subject of simulation: the real-world biological system, and the relevant information known about it.

The components are:

- draw an explanatory *Cartoon*
- provide an *overview* description of the domain
- provide a *Glossary* of relevant domain-specific terminology
- Document *Assumptions* relevant to the domain
- define the *scope and boundary* of the domain – what is inside and what is outside
- identify relevant *sources*: people, literature, data, models, etc

Discovery > Domain > Cartoon

CoSMoS pattern: *Cartoon:* Sketch an informal overview picture of the *Domain*.

Figure 1 is a cartoon of the regulatory process. A single gene regulation and its expression is conceptually relatively straightforward; the complex interplay of multiple interacting regulatory processes is not.

Discovery > Domain > overview: embryonic stem (ES) cell biology

Modern, high-throughput laboratory techniques routinely provide large-scale datasets including complete genome sequences, dynamic measurements of gene expression, extensive lists of regulatory proteins and RNAs, and *in vivo* occupancy of DNA by TFs, cofactors and nucleosomes [7]. Such datasets facilitate the investigation of ES cell regulatory networks. To create a complete multi-layered model of a stem cell network one should exploit these big data to bridge gaps between the phenotypic behaviour of whole cells and key regulatory molecules [50].

We need to capture the results of multiple high-throughput experiments within a logical and transparent conceptual and computational framework in order to facilitate the interrogation of multiple layers of complex regulatory information. Our initial model is based on the complete genome sequence of mouse embryonic stem cells and on ChIP-Seq data that capture the density of TF binding sites throughout the genome. TFs operate in parallel, influencing each other; according to our hypothesis, they produce genome-wide interference patterns that capture in some way the predicted nature of the entire pluripotent circuitry.

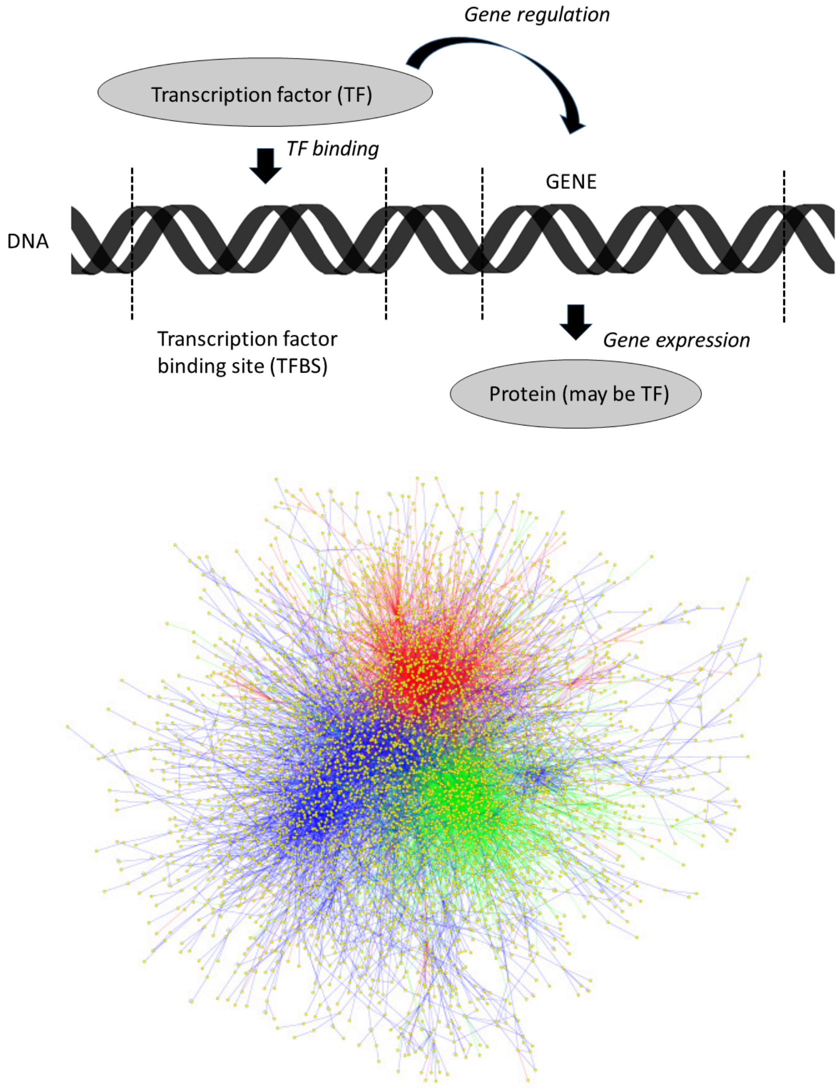


Fig. 1. Domain > Cartoon: (top) The regulatory process: a TF protein binds to DNA at the BS, thereby regulating production of protein (which may be a TF) from the corresponding gene (gene expression). (bottom) Expressed proteins may include other TFs that can regulate expression of other genes: a 'hairball graph' of the human proteome and its binding interactions [20, fig.1]

Embryonic stem (ES) cells have the potential to produce all of the different cell types within the body, but this behaviour cannot yet be efficiently exploited *in vitro*. We have considerable knowledge of the component parts of the regulation of ES cells maintained under precise external conditions [35], but during normal development many different types of regulatory factors interact, enabling cells to respond flexibly to changing environments. The regulatory network of single ES cells is therefore some function of both cell intrinsic and cell extrinsic variables.

Here we assume that pluripotency is a state of individual ES cells. ES cells exit pluripotency via a transient 'primed' state that facilitates cell fate computation [38]. Our knowledge of this exit process and the transient primed state is incomplete, partly because it is difficult to obtain data from transient cell states [47]. The process of pluripotency exit itself is intrinsically disorganised and/or chaotic in order for it to integrate intrinsic and extrinsic information and compute cell fate. According to our conceptual framework, regulatory circuitries compute cell fate trajectories via 'critical-like dynamics' at the edge of chaos [24].

Nanog, Oct4 and Sox2 form part of the core pluripotency circuitry of ES cells [13]. Oct4 in particular seems central to understanding pluripotency. Oct4 expression level is closely regulated, with deviations either above or below a certain expression range resulting in differentiation [40]. It has been suggested that protein complexes, in which Oct4 is involved, help to establish a dynamic competition between individual elements, serving to buffer the differentiation-promoting activity of Oct4 [37].

Fluctuations are inevitable in any system that has many degrees of freedom. At static equilibrium, such fluctuations ultimately disappear but under non-equilibrium conditions, fluctuations are often great enough to drive reorganisation toward new dynamic states [15, 39]. If continual driving is experienced, complex spatiotemporal patterning usually results and systems are said to have 'self-organised' [11, 22, 39].

In biology, the growth and development of organisms occurs far from equilibrium. The stem cell regulatory networks that facilitate these processes are replete with positive and negative feedback loops and nonlinear interactions. When faced with overwhelming complexity, the natural tendency of humans is to either reduce, simplify or ignore it. Reductionist thinking makes systems (a) easier to think about, (b) easier to consider manipulating, and (c) easier to predict, provided non-equilibrium driving is minimal.

Over the last few decades, there has been increasing awareness of the limitations of the reductionist approach [8, 16, 19, 30, 41] and it has become clear that some laws of nature cannot be deduced by resolving more detail [48]. This so called 'new era of physics' focuses on developing complex behaviour out of simplicity, instead of the traditional reductionist approach that reduced complexity to its simplest possible form [2, 28, 41]. Non-equilibrium driving can have profound consequences on system behaviour, a realisation that contrasts with our natural tendency to assume systems are near equilibrium or at least show some steady state behaviour. Equilibrium and reductionist thinking pervades most scientific disciplines [9–11, 18], including molecular and stem cell biology.

The differentiation of pluripotent cells in the early embryo is a fascinating non-equilibrium process that results in the production of numerous specialised cell types. More than 600 different proteins have been implicated in exit from a naïve pluripotent state and control of early state transitions in the mouse [29]. As our focus shifts from individual components to complex communication networks, experimental studies have become more difficult. Not only do central features of complex networks, such as robustness, prevent straight forward analysis and interpretation of network behaviours, but many experiments cannot be performed because of ethical reasons surrounding the use of human embryos.

Computer simulation sidesteps the ethical, moral and political issues surrounding use of human embryos. It therefore represents an alternative route to gaining new insight in to this promising field of regenerative medicine. Our overarching aim is to gain sufficient understanding so that any cell type of therapeutic interest can be generated effectively at will.

Discovery > Domain > Glossary: terms and acronyms

> CoSMoS pattern: *Glossary:* Provide a common terminology across the simulation project.

The main biological terms used in the various models are:

binding site (BS) : section of DNA that binds a given TF and influences transcription of associated genes

branching process (BP) : the mathematical model underlying inspiration of the TF BP framework being investigated here

ChIP-Seq : a technique to identify the binding sites of transcription factors on DNA

cistrome : the portion of the genome associated with a specific TF; a pattern of genome-wide binding sites to which the TF displays some activity

pluripotent stem cell : a cell capable of generating all the cell types present in the adult body

segment : the genome data is segmented, into say 10k or 50k base-pair sequences, in order to apply the TFBP framework

transcription factor (TF) : a protein that binds to DNA to influence transcription of the associated gene

Discovery > Domain > assumptions

See §3.1 for the *Assumptions* pattern requirements.

A.4 The genome can be modelled as a set of overlapping TF cistromes without needing epigenetic factors
  reason We are looking only at TF segments, and the pluripotent state can be induced by TFs alone
  justification See, for example, [31]
  consequence Behaviours facilitated by other factors, such as epigenetics, will be unseen in the model

A.5 a TFBS is either bound or unbound, there is no partial TF binding
  reason not enough data to say otherwise

A.6 a segment can be either activated or deactivated, there are no differing amounts of activation
  reason Simplification: the data does say whether a segment has one or more binding sites
  justification This is the first iteration; we will revisit the necessity/impact of this assumption in later iterations
  consequence We will not be able to separate out behaviours of groups of genes in a segment. In order to do so, we could use smaller segments. But segments cannot be made too small, else we would lose correlations between related TFs.

A.7 we can investigate cell decision making by modelling an individual cell, not a population
  reason cells have internal decision making, although they can also be influenced by their environment
  justification See, for example, [33]
  consequence We will not be able to investigate population-level decision making

Discovery > Domain > scope

- single cell model
- single transcription factor model
- later iterations will add more, coupled TFs, and more interacting cells

Discovery > Domain > sources

- Domain scientists
- Biological literature, as referenced in the various overviews
- Chip-seq data for various cistromes (source: Dietmann)

3.3 Discovery > Domain Modelling

CoSMoS pattern: *Domain Modelling:* Produce an explicit description of the relevant domain concepts.

The components of *Domain Modelling* are:

- *collaborate* with the identified Domain Scientist
- draw an explanatory *Cartoon*
- discuss and choose the *Modelling Approach* and level of abstraction
- build the *Domain Model* using the chosen modelling approach
- build the *Data Dictionary*
- document *Assumptions* relevant to the domain model
- Argue Domain Model Appropriate (omitted here)

Discovery > Domain Modelling > collaborate

The lead domain scientist (Halley) and the domain modellers (Greaves, Stepney) collaborated closely throughout the development of the domain model, translating and abstracting the conceptual TF BP model into a form suitable for simulation.

The domain scientists (Halley, Smith, Dietmann) collaborated on refining the research context.

The simulation engineer (Greaves) collaborated with the the data collection expert (Dietmann) on the form and content of the biological data provided.

Discovery > Domain Modelling > Cartoon

See §3.2 for the *Cartoon* pattern.

Due to the structure of our Domain Model description, the *Domain Modelling Cartoon* is presented in the section on the TF BP model (figure 4), and should be read in in that context.

Discovery > Domain Modelling > Modelling Approach

> CoSMoS pattern: *Modelling Approach:* Choose an appropriate modelling approach and notation.

A central part of this design process is to develop the simplest possible working model at each stage of the modelling process. This 'agile' approach ensures that simulation code is not unnecessarily complicated. It also helps to ensure that if a coding problem is found, it is simple matter to backtrack to the last working model.

The domain model is captured using UML, in anticipation of an agent-based, object-oriented design and implementation of the simulator.

Discovery > Domain Modelling > Domain Model

Our domain modelling gives rise to several models at different levels of abstraction: a specifically biological stem cell model of regulatory networks, a model simplifying detailed transcription regulatory networks using branching process theory, and a generic abstract model, which we refer to as the 'sparking posts' model.

Note that the sparking posts model could also be used as a domain model for other biological phenomena as captured by branching process theory, such as patterns of information flow in the human brain.

Regulatory network

We have mouse genome data including the suite of BSs within it. For convenience and simplicity, we divide this sequence in to 50 kilobase (kb) segments, any of which may or may not contain binding sites for a particular TF of interest. If a 50kb segment contains a binding site for our transcription factor, X, then the segment is said to be part of the X cistrome.

Data about the locations of the transcription factor binding sites, in relation to the gene segments in the model, is provided experimentally by ChIP-Seq data. Figure 2 is a representation of ChIP-Seq data.

The regulatory network components can be captured in a model such as that shown in figure 3. However, we abstract away from many of these 'hairball' inducing details, and consider the system instead in terms of the TF BP model.

Transcription Factor Branching Process model

A common approach to understanding cell regulatory processes is the

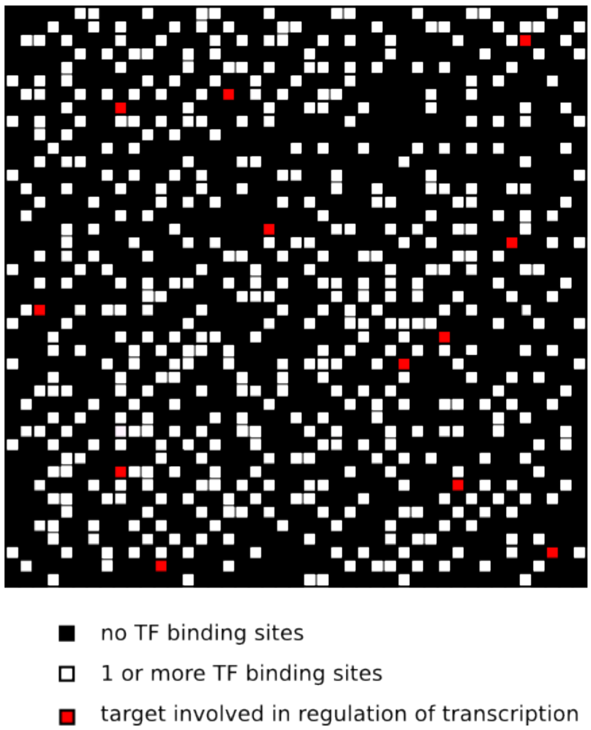

■ no TF binding sites
□ 1 or more TF binding sites
🟥 target involved in regulation of transcription

Fig. 2. A representation of a set of ChIP-Seq data for a cistrome (part of the genome relevant to a specific TF). Each square represents a 50kb segment of DNA. A white square is a segment that contains at least one BS site for a product that is not a TF. A red squares is a segment that contains at least one BS site for a product that is a TF. A black square is a segment that does not belong to this cistrome.

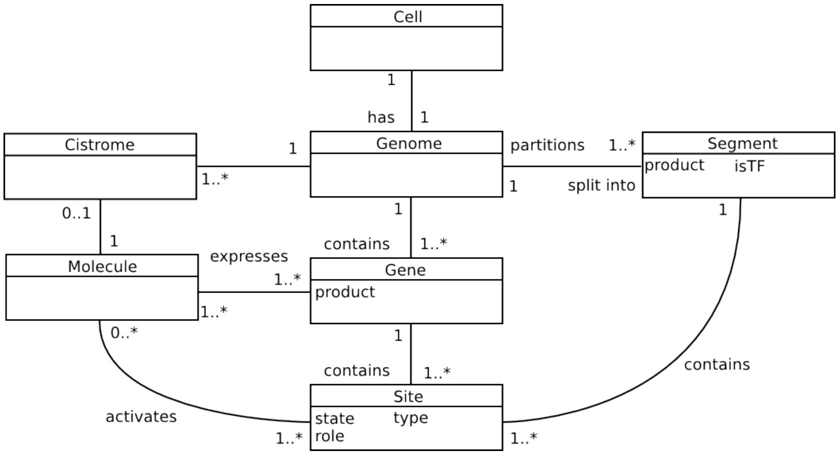


Fig. 3. Class diagram model of a stem cell pluripotency regulatory network. The stem cell has a genome comprised of genes, which can alternatively be described as a cistrome (or set of cistromes), each being comprised of segments of gene which may or may not contain transcription factor binding sites.

application of concepts, tools and techniques developed in mathematics, physics or computer science [34]. Network representations, for example, can accommodate multiple types of data within a single visual illustration that provides an overview of regulatory pathways and components [21, 34]. As already mentioned, empirically-derived interaction networks can be difficult to interpret, often appearing as a 'hairball' graph as regulatory mechanisms are increasingly dissected.

We use here a novel way to visualise and simulate genome-wide regulatory network interactions. Our coarse-grained approach does not require details of binding constants prerequisite for most ODE models of stem cell regulation. In many previous computational or mathematical models of stem cell regulatory networks, TFs are represented as single nodes with binary (*on*/*off*) behaviour. Here, we use a different approach that captures TF activity as a dissipative branching process that propagates within the bounds imposed by the TF's unique cistrome.

Unlike reductionist models that capture TF activity using single variables in an equation, in our model we explicitly represent a background delocalisation of TF activity throughout the genome. We can visualise the activity of each TF's BP as a kind of gateway through which regulatory information pertaining to the TF passes over time.

The TF BP model allows a decoupling between details of BS constants and the emergent effect of TF activity throughout the genome.

Instead of struggling with countless (often unknown) binding constants, we consider the overall flow of regulatory information at genome-wide scales. It is thus more suitable for attempts to discover how the ES cell regulatory network behaves as a whole during computation of lineage choice. Through this more coarse-grained methodology, we hope to discover complex interactions that can easily be overlooked by studies that focus on only a handful of key regulatory components at a time.

The potential binding of a TF to target regions throughout the genome is determined by ChIP-Sequencing. The data set or 'footprint' for a given TF comprises a unique pattern of TF-DNA interactions that is somewhat dependent upon the precise methods used to infer interactions. The precise footprint for a specific TF may vary between different experimental datasets. Such 'fuzziness', rather than being a nuisance, is intrinsic to the TF BP model.

If we understand the activity of any given TF as a branching process of regulatory information propagating through time, it makes sense that there will be some correlation between observed TF expression and the saturation of target sites influenced by TF activity. The significance of this important point will become clearer in later work, when we simulate multiple cistrome data sets. Here, we focus on simulating a single TF's BP to introduce the groundwork for our approach.

Figure 4 presents a Cartoon of the TF BP model. Each square in the figure corresponds to a 50kb segment of the mouse genome. Black squares represent segments that contain no BSs for the TF of interest, while red and white squares represent segments with at least one BS for the TF of interest. The difference between a red and white segment lies in their products. A red segment has products that include TFs, whereas none of the products of a white segment is a TF. Henceforth, when we refer to a 'red' segment we mean a gene segment that can bind TF and thus become stimulated into transcribing further TFs.

We capture the countless (ill-defined or unknown) cascades of gene activation via TF production and feedback as a branching process in which TFs produce other TFs while also regulating the remainder of the genome. There are potentially three qualitatively different types of behaviour for any TF_X branching process. Firstly, the cistrome X is saturated and the TF_X gene is continually and stably expressed. Alternatively, there is the opposite type of emergent behaviour, with TF_X expression occurring at a very low noisy level that is not sustainable unless TF_X is supported by continual activation of the TF_X gene via some external signal. Finally there is a dynamic intermediate between these extremes where a branching process only just percolates through the TF_X cistrome. In all cases, the targets of TF_X are divided in to two

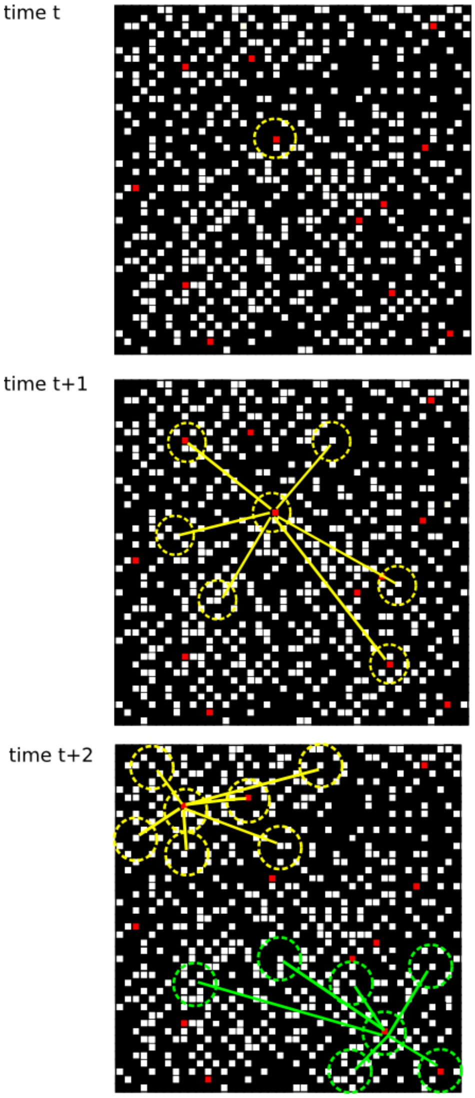


Fig. 4. Domain Modelling > Cartoon: A branching process representation of the overall flow of regulatory information, which serves as the basis of our simulation. At t, assume the circled red segment is activated. At time $t + 1$ this will activate m further randomly chosen segments (arrows), and itself deactivate. At time $t + 2$, any of these newly activated segments that are themselves red, will each activate a further m randomly chosen segments, and deactivate.

types: (1) dissipative targets that do not propagate information back in to the TF_{X} cistrome and (2) amplifying targets that are either TFs themselves and capable of propagating information or code for signalling molecules that are involved in signal transduction.

We define an average branching ratio, called m, for our gene regulation branching process. That is to say that once transcribed, a gene (or gene segment in our case) will produce m product molecules (in this single cistrome model these will all be the TF that binds to binding sites within the cistrome of interest). If the activated site is associated with TF products then new TFs are produced and these can bind to other TF binding sites in the system. In this way, up to m segments will be activated in the next time step of the algorithm. In the time step after this each of the activated segments can go on to activate m further segments and so on as illustrated in Figure 4.

This TF BP model is built on the classical BP theory outlined in section 'Domain > overview', and is adapted in the following ways:

- m is related to the BP branching factor μ, but is not the same, because here the m 'offspring' include both white and red segments, yet only red segments go on to produce further 'offspring'.
- In the supercritical case, the number of offspring cannot increase without bound, but only up to the number of relevant segments in the cistrome.
- The individulas are segments, and do not 'die' at the end of a generation; rather they can be reused (reselected) in subsequent generations.

Domain Model: Sparking Posts

In order to model a branching process, we produce our domain model in terms of a metaphor. To capture the nature of critical-like self-organisation hypothesised to underpin lineage computation, we have reduced the system to a 'sparking posts model'. This computational model is used to define the backbone of critical-like self-organisation upon which other layers of complexity are elaborated.

The TF BP representation of our system is modelled as a 'sparking posts' representation of the cistrome in which each segment is modelled as a metal 'post' which emits 'sparks' once it has been activated by an incoming spark emitted by another post in the previous timestep. The sparks represent the TF products of the genes contained within a given segment and are therefore the principal mode of communication between cistromes, the genome being effectively the sum of all cistromes in the system.

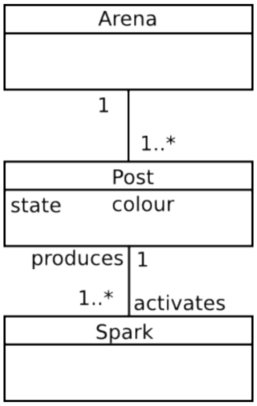


Fig. 5. A class diagram capturing the sparking posts model components. An Arena has multiple Posts; Posts produce Sparks, and are activated by Sparks.

So the Domain Model is as follows.

Consider an *arena* containing metal *posts*, some *red*, some *white*. The arena is an abstraction of a particular cistrome; the posts are abstractions of the segments containing BSs (red and white squares in figure 2); red posts are abstractions of segments that express TFs (red squares in figure 2).

Posts may be *active* (on) or not. In a timestep, an active red post emits *m sparks*. A post being active is an abstraction of a gene in a segment being activated; a red post sparking is an abstraction of an activated gene expressing a TF.

Posts become deactivated after they have sparked. A spark lands on a random post in the arena (that is, the model is aspatial), and activates it.

Continued propagation of sparks relies on the activation of sufficient red posts at each timestep.

Figures 5 and 6 capture this Domain Model.

Discovery > Domain Modelling > Data Dictionary

> CoSMoS pattern: *Data Dictionary:* Define the modelling data used to build the simulation, and the experimental data that is produced by domain experiments and the corresponding simulation experiments.

The sparking post model's parameters and variables are shown in figure 7. Figure 8 shows the values of some of these parameters for the cistromes of interest here.

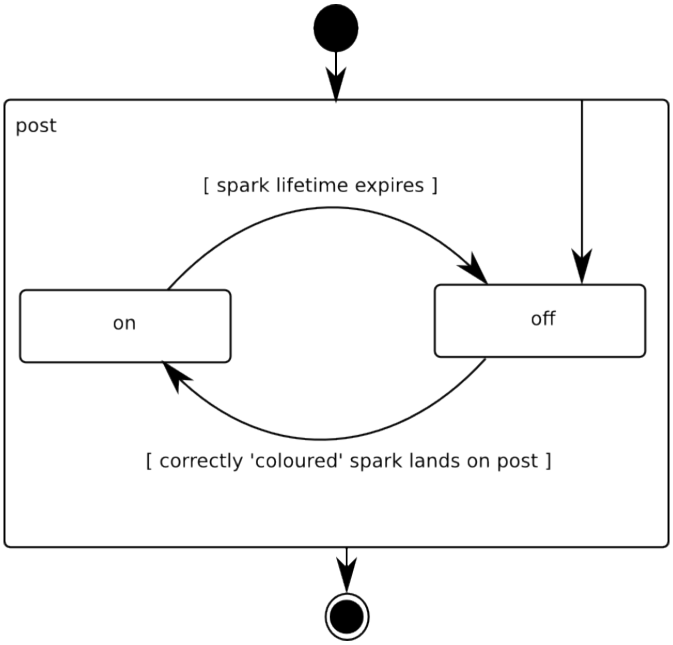

post
[spark lifetime expires]
on
off
[correctly 'coloured' spark lands on post]

Fig. 6. A state diagram of a post. Posts are initially off; become activated (on) if a spark lands; then become deactivated in the next timestep.

p	total number of posts in the arena
r	number of red posts
m	sparks emitted per active red post
s_0	number of red posts active initially
t	timestep
s_t	number of red posts active at timestep t

Fig. 7. Sparking post model: (top) parameters, constant during a simulation run; (bottom) variables, changing during a simulation run

	Nanog	Sox2	Oct4
p	4310	3330	2540
r	631	542	466
r/p	0.146	0.163	0.183
$p/r = m_c$	6.8	6.1	5.5

Fig. 8. The values of the parameters p (number of posts, or segments in the cistrome) and r (the number of red posts, or red segments in the cistrome) for the TFs investigated in this study

Discovery > Domain Modelling > assumptions

See §3.1 for the *Assumptions* pattern requirements.

First, we have some assumptions related to the TF BP model, which we note as they have an impact on the sparking posts model.

A.8 the product of a TF producing segment is the TF whose cistrome we are modelling

reason An assumption underlying use of the TF BP model

justification The TF may not be directly produced; there may be a cascade of production, but the TF BP model collapses this cascade. We are investigating this model.

consequence This is an abstraction from the biology, made to allow us to model the highly complex processes. If it works, this abstraction could also provide an approach to include other features such as epigenetics and mRNAs in a tractable model.

A.9 the identity of the TFs produced during transcription is irrelevant in the single cistrome model

reason An assumption underlying use of the TF BP model

justification The TF BP model assumes that the relevant scale of computation is the cistrome level, abstracted from specific details of the individual TFs

Assumptions directly related to the sparking posts model are:

A.10 a spark from a post can hit any post with equal probability: there is no notion of a 'distance' between posts

reason an aspatial model

justification the TF BP model collapses a potential cascade of TFs into a single 'proxy' TF. This cascade would lose any spatial dependence in the DNA.

A.11 a post cannot be hit by more than one spark per timestep: there is no notion of different 'capacity' posts

reason follows from assumption A.6

Discovery > Domain Modelling > Expected Behaviours

CoSMoS pattern: *Expected Behaviours:* Describe the expected emergent behaviours of the underlying system.

The 'sparking posts' domain model forms the basis for subsequent simulation development.

We can form a much simpler version of the model, in order to help understand the effect of noise. Since there are a finite number of posts, stochastic fluctuations will occur, and sparks might occasionally miss many or all of the red posts. Here we instead assume that posts are always hit the average number of times. We are interested in the proportion of red posts active in the 'steady state', in limit of large time.

At time t there are s_t red posts active. Each of these active post emits m sparks, so a total of $s_t \times m$ sparks are emitted. Let each of these sparks be absorbed by a separate post, of which a fraction r/p are red. So at the next timestep, there are $s_{t+1} = s_t mr/p$ red posts active.

The number of active red posts reduces with time if $m < p/r$, and so the arena is extinguished, with $s_\infty = 0$.

The number of active red posts steadily grows with time if $p/r < m$, until there are more sparks emitted than there are posts in total (moving outside our assumption of each spark being absorbed by a separate post), and so the arena saturates with $s_\infty = r$.

The critical value, m_c, where this change of behaviour happens is $m_c = p/r$. Values for m_c for the TFs of interest are shown in figure 8.

Hence the expected behaviour of the single cistrome simulation is to quench for low values of m, saturate for high values of m, and have a tipping point around m_c.

4 Development phase

CoSMoS pattern: *Development:* Build the scientific instrument: produce a simulation platform to perform repeated simulation, based on the output of the *Discovery* phase.

The components of the development phase are:

- *revisit* the Research Context
- develop a *Platform Model*
- develop a *Simulation Platform*
- Argue Instrument Built Appropriately (omitted here)

4.1 Development > revisit

The research context is unchanged in the light of *Discovery* phase activities. The TF concepts need to be reinterpreted in terms of the sparking posts model.

4.2 Development > Platform Modelling

> CoSMoS pattern: *Platform Modelling:* From the *Domain Model*, develop a platform model suitable to form the requirements specification for the *Simulation Platform*.
>
> The relevant components of *Platform modelling* are:
>
> – choose a *Modelling Approach* for the platform modelling
> – develop the *Platform Model* from the Domain Model
> – document *Assumptions* relevant to the platform model

Development > Platform Modelling > Modelling Approach

We use the same approach as for domain model, assisting seamless development.

Development > Platform Modelling > Platform Model

The emergent tipping point behaviour is not part of the platform model. The rest of the 'sparking posts' model carries over from the domain model unchanged.

Instrumentation is added, to collect statistics from the simulator, including post sparking activity. A user interface and visualisation component is added, to control the simulator runs (set the simulation parameters), and examine the output.

Development > Platform Modelling > Assumptions

A.12 the sparks due to an activated post last for one simulation time step
reason simplicity
justification first iteration
consequence half lives and decay rates are not modelled; they may be added in later iterations

4.3 Development > Simulation Platform

CoSMoS pattern: *Simulation Platform:* Develop the executable simulation platform that can be used to run the *Simulation Experiment*.

The relevant components of developing the simulation platform are:

- choose an Implementation Approach
- code and test (details omitted here)
- perform calibration (details omitted here)
- document *Assumptions* relevant to the simulation platform

Development > Simulation Platform > implementation approach

The simulation is implemented as an object-oriented Java application using the MASON simulation environment to handle such things as time-stepping the simulation and on screen graphics (when running in graphical mode).

5 Exploration phase

CoSMoS pattern: *Exploration:* Use the simulation platform resulting from *Development* to explore the scientific questions established during *Discovery*.

The components are:

- *revisit* the Research Context
- perform *Results Modelling*
- perform a *Simulation Experiment*
- Argue Instrument Used Appropriately (omitted here)

5.1 Exploration > revisit

The research context is unchanged in the light of *Discovery* and *Development* phase activities.

5.2 Exploration > Results Modelling

CoSMoS pattern: *Results Modelling:* Develop a results model suitable for interpreting simulation experiment data in Domain Model terms.

The relevant components of results modelling are:

- build a *Visualisation Model*
- build a *Results Model*

– Argue Results Model Appropriate and Consistent (omitted here)

Exploration > Results Modelling > Visualisation Model

CoSMoS pattern: *Visualisation Model:* Visualise the simulation experiment results of the *Data Dictionary* in a manner relevant to the users.

The visualisation mimics the cistrome data in figure 2.

Exploration > Results Modelling > Results Model

The results model is the cistrome activity (number of activated posts) as a function of time.

5.3 Exploration > Simulation Experiment

CoSMoS pattern: *Simulation Experiment:* Use the simulation as a scientific instrument to explore the behaviour of the system.

The relevant components of a simulation experiment are:

- design the experiment
- perform the experiment
- analyse the results

Exploration > Simulation Experiment > design

The parameters p (number of posts) and r (number of red posts) are effectively fixed for any given set of experimentally derived cistrome data (figure 8). We can also generate synthetic data to create systems with a range of p and r values to explore general behaviours.

We identify 4 experiments to perform on the single-arena simulation:

experiment 0 : Effect of m. With p and r fixed and $s_0 = r$, explore the effect of m by locating those values of m for which the system remains fully saturated: all red posts are activated at all time steps. Compare this with the expected m_c value (figure 8) for a noiseless system.

experiment 1 : Effect of s_0, sensitivity to initial conditions. Repeat experiment 0 with smaller values of s_0.

experiment 2 : Effect of r. Create arenas with a fixed p and a range of r values. At each value of r, determine the values of m for which the system remains saturated throughout the simulation.

experiment 3 : Effect of noise. Keeping the ratio of p to r fixed at the value in the biological data, investigate the effect of reducing p. This will give some insight into how the data scales up within the context of our model, and whether we can use smaller arenas in experiments to improve simulation performance.

Number of simulation runs. We are not performing any statistical analyses at this stage of the project, merely inspecting behaviour. However, the simulation is essentially stochastic, and when we do come to perform statistics, we will need to choose the number of runs based on the significance, power, and effect size of interest. For consistency, we make that choice now, and use the relevant number of runs.

We require a statistical significance of 99% (a 1% false positive rate), a statistical power of 99% (a 1% false negative rate), and a 'medium' effect size (Cohen's $d = 0.5$, the ability to distinguish a difference in means of 0.5 of a standard deviation). Calculating the required sample size for these experimental parameters[3] gives 192.

We round this up, and take the number of runs to be $N = 200$.

Protocol. One simulation run comprises the p and r values of a particular arena (chosen to match Nanog, Sox2, Oct4 data), an m value (1–50), and a starting activity ($s_0 = r$ for experiment 0; $s_0 = r/2$ for experiment 1).

For each simulation run, we record the proportion of active red posts at the final timestep, $T = 1000$.

For each parameter set (p, r, m, s_0), we run the simulation $N = 200$ times.

5.4 Exploration > Simulation Experiment > analyse results

Experiments 0 and 1

Experiment 0 uses $s_0 = r$: all red posts initially active. Experiment 1 uses $s_0 = r/2$: half the red posts initially active.

See figures 9–11 for the results of the simulation runs.

The observed values of m where the system 'switches on', and can maintain saturation, are close to the calculated m_c values. However, m has to be somewhat higher than this to saturate the finite-sized arena.

Starting with only half the posts active makes little difference to the results.

[3] using, for example, the calculator at http://powerandsamplesize.com/Calculators/Compare-2-Means/2-Sample-Equality

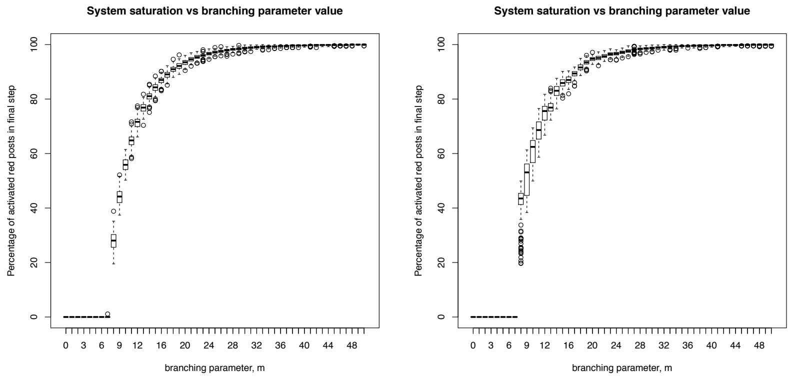


Fig. 9. p and r corresponding to Nanog data; (left) experiment 0: $s_0 = r$; (right) experiment 1: $s_0 = r/2$. Recall $m_c = 6.8$

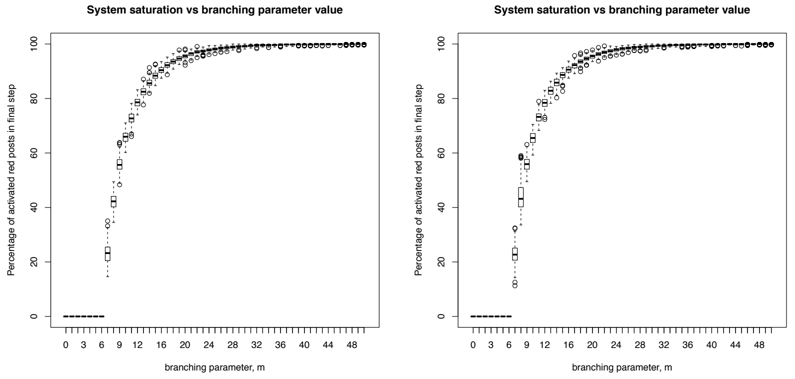


Fig. 10. p and r corresponding to Sox2 data; (left) experiment 0: $s_0 = r$; (right) experiment 1: $s_0 = r/2$. Recall $m_c = 6.1$

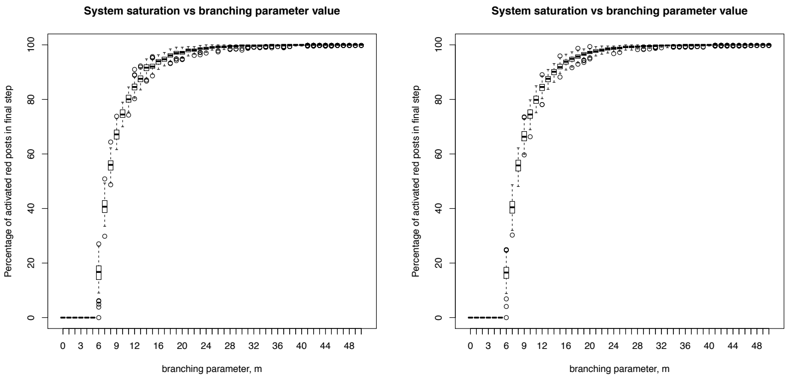


Fig. 11. p and r corresponding to Oct4 data; (left) experiment 0: $s_0 = r$; (right) experiment 1: $s_0 = r/2$. Recall $m_c = 5.5$

Experiment 2

For experiment 2, we took $p = 4310$ (as in Nanog), and $r = 200, 400, 600, 800$, to see how the value of m_c changes. We used $s_0 = r$ throughout.

See figures 12–13 for the results of the simulation runs.

Recall that the theoretical tiping point value is $m_c = p/r$. So as r increases, m_c should decrease. This is observed (figure 12).

Also note that the smaller r, the noisier the behaviour. This is expected as stochastic effects will be more prominent when there are fewer red posts available.

Experiment 3

For experiment 3, we took $p/r = 4310/631$ (as in Nanog), and reduced p keeping p/r constant (mimicking a smaller arena but with the same density of red posts). We used $s_0 = r$ throughout.

See figure 14 for the results of the simulation runs; compare with figure 9(top) for the ‘full’ arena.

The systems tip at the same point, but the behaviour gets noisier as p (and hence r) decreases, and stochastic effects become more pronounced.

6 Discussion

This paper documents and illustrates the use of CoSMoS patterns to perform a complete iteration of a CoSMoS simulation project, from ini-

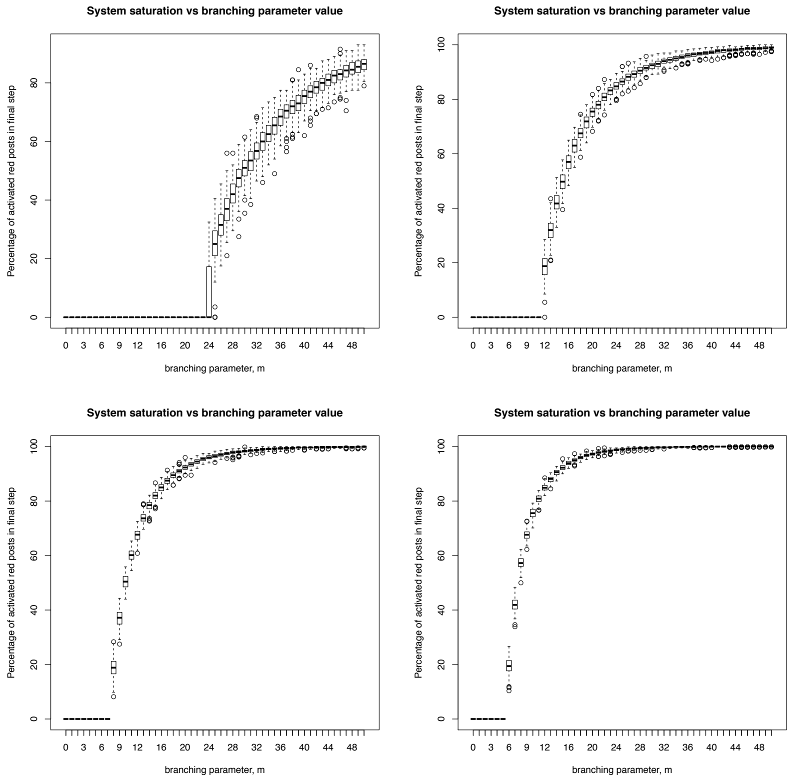


Fig. 12. Experiment 2: varying r; here $p = 4310$: (top left) $r = 200$; (top right) $r = 400$; (bottom left) $r = 600$; (bottom right) $r = 800$

r	m obs	m_c
200	23–24	21.6
400	11-12	10.8
600	7–8	7.2
800	5–6	5.4

Fig. 13. Experient 2: observed value of m at tipping point, versus calculated value m_c

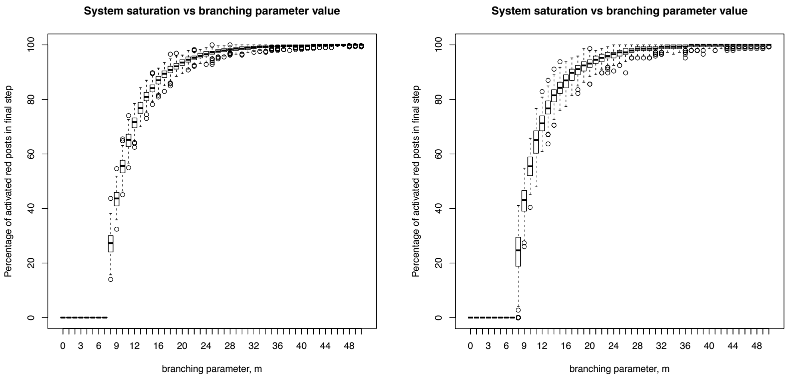


Fig. 14. Experiment 3: varying p with constant p/r: (left) $p = 2000, r = 293$; (right) $p = 1000, r = 146$

tial discovery, through development, to exploration. There were several lessons learned, summarised here.

It is not always clear whether information should be included in the Domain, or Domain Model, sections, particularly relating to assumptions. However, it is more important to document the information that to agonise over precisely which section to document it in.

Not all patterns are applicable. For example, here the Domain Model Cartoon had to be presented within the Domain Model section, rather than as a prior illustration. Additionally, the TF BP model is so abstracted from the Domain, that aspects such as the Domain Experiment Model [5] are not relevant, and so have been omitted. Again, it is more important to follow the spirit of the CoSMoS approach rather than the letter of every pattern.

Not every aspect of the CoSMoS approach needs to be performed with complete rigour. This simulation is not safety critical, so some aspects have been omitted (such as justification of all assumptions, and argumentation of fitness-for-purpose). The extra effort needed to complete all aspects should be expended only if it gives benefit.

Although the presentation is sequential and hierarchical, the historical process was not. We spent many short iterations, and considerable backtracking (for example, see figure 3), before finally fixing on the ‘sparking posts’ model. The CoSMoS patterns define what information should be recorded by the end of the project, but not the order it needs to be produced. Some uses of CoSMoS can apply the patterns in significantly different orders, for example [4].

We might not have arrived at the conceptual sparking posts model without taking an iterative approach. The need to have just a single-cistrome model for this first iteration revealed a fundamental misunderstanding that the modellers were having about the background TF BP model.

Although we were taking an agile approach, producing minimal simulation models and code, collaborations meetings would often generate interesting but out of current scope ideas. We invented the concept of the "to don't" list: a place to record the ideas for future reference, in a manner that made it clear they were not to be included in the current iteration. Some of these ideas also prompted the recognition of assumptions in the current iteration.

The Domain Scientist (Halley) was new to the CoSMoS approach at the start of the project, but had previous experience working with modellers using different approaches on other projects. Halley reports that CoSMoS is a flexible tool to produce objective scientific simulations, and allows progress without being funnelled into preconceptions imposed by a specific toolset or implementation approach.

7 Summary, Conclusions, Future Work

This work has run through a complete CoSMoS cycle, producing the first iteration of the system: a single cistrome model.

The results demonstrate that the single-cistrome model exhibits its tipping point close to the predicted value of m_c, but the tipping is not particularly sharp, so for values of m close to m_c, there is a lot of noise in the system.

In order to generate results that have genuine biological relevance, it will be necessary to create a simulation of two or more cistromes interacting with each other via the TFs that each produces. For example, we will investigate model behaviour when the Oct4, Sox2 and Nanog branching processes are allowed to interact. Given the groundwork developed in this first iteration, the modelling and simulation work in for the second iteration, to augment the system with multiple cistromes, should be relatively straightforward. We are currently developing this second iteration.

Beyond this, future iterations could include:

– More complex connections within networks of cistromes, including inhibition and negative feedback, combinatorial binding of TFs, and indicators of 3D genomic or chromosomal architecture. The inclusion of inhibition of gene expression is particularly relevant to the process

of pluripotency exit, as batteries of differentiation genes are suddenly expressed.
- A Domain Specific Language with which we can describe the network
- TF half life variability
- Epigenetic histone marks that may help to shape circuitry self-organisation
- Combinatorial binding of TFs to enhancer sites that impart transcriptional synergy [46]
- Multicellular model incorporating cell-cell signalling

The model presented here represents a novel example of self-organisation that may apply to other complex systems. It is of interest from a purely theoretical perspective because it helps to demonstrate how distributed interactions among units result in higher ordered emergent behaviours. Such complexity could provide dynamic templates of organisation upon which natural selection builds additional elaborations [25].

Acknowledgments

This work was performed as part of the CellBranch project, funded by the UK's Biotechnology and Biological Sciences Research Council (BB-SRC), project reference BB/L018705/1.

References

[1] C. Alexander *et al.*, *A Pattern Language: towns, buildings, construction.* Oxford University Press, 1977.

[2] P. W. Anderson, "Is complexity physics? is it science? what is it?" *Physics Today*, vol. 44, no. 7, p. 9, 1991.

[3] P. S. Andrews, F. A. C. Polack, A. T. Sampson, S. Stepney, and J. Timmis, "The CoSMoS process, version 0.1: A process for the modelling and simulation of complex systems," Department of Computer Science, University of York, Tech. Rep. YCS-2010-453, Mar. 2010.

[4] P. S. Andrews and S. Stepney, "Using CoSMoS to reverse engineer a domain model for Aevol," in *Proceedings of the 2014 Workshop on Complex Systems Modelling and Simulation, New York, USA, July 2014*. Luniver Press, 2014, pp. 61–79.

[5] ——, "The CoSMoS Domain Experiment Model," in *Proceedings of the 2015 Workshop on Complex Systems Modelling and Simulation, York, UK, July 2015*. Luniver Press, 2015, pp. 1–8.

[6] P. S. Andrews, S. Stepney, and J. Timmis, "Simulation as a scientific instrument," in *Proceedings of the 2012 Workshop on Complex Systems Modelling and Simulation, Orleans, France, September 2012*, S. Stepney, P. S. Andrews, and M. Read, Eds. Luniver Press, 2012, pp. 1–10.

[7] A. Ay and D. N. Arnosti, “Mathematical modeling of gene expression: a guide for the perplexed biologist,” *Critical Reviews in Biochemistry and Molecular Biology*, vol. 46, no. 2, pp. 137–151, 2011.

[8] P. Bak and M. Paczuski, “Why nature is complex,” *Physics World*, vol. 6, no. 12, pp. 39–43, 1993.

[9] ——, “Complexity, contingency, and criticality,” *Proceedings of the National Academy of Science USA*, vol. 92, pp. 6689–6696, 1995.

[10] P. Ball, “Transitions still to be made,” *Nature*, vol. 402, pp. C73–C76, 1999.

[11] ——, *The Self-Made Tapestry.* Oxford University Press, 2001.

[12] S. Bornholdt, “Less is more in modeling large genetic networks,” *Science*, vol. 310, pp. 449–451, 2005.

[13] L. A. Boyer, T. I. Lee, M. F. Cole, S. E. Johnstone, S. S. Levine, J. P. Zucker, M. G. Guenther, R. M. Kumar, H. L. Murray, R. G. Jenner, D. K. Gifford, D. A. Melton, R. Jaenisch, and R. A. Young, “Core transcriptional regulatory circuitry in human embryonic stem cells,” *Cell*, vol. 122, pp. 947–956, 2005.

[14] D. Bray, “Molecular networks: the top-down view,” *Science*, vol. 301, pp. 1864–1865, 2003.

[15] E. Chaisson, “Complexity: An energetics agenda,” *Complexity*, vol. 9, no. 3, pp. 14–21, 2004.

[16] J. P. Crutchfield, J. D. Farmer, N. H. Packard, and R. S. Shaw, “Chaos,” *Scientific American*, vol. 255, no. 6, pp. 38–49, 1986.

[17] M. E. Driscoll, “Is big data at a tipping point?” http://www.analyticbridge.com/profiles/blogs/is-big-data-at-a-tipping-point, 2009, [Accessed: 2015-05-01].

[18] I. Ekeland, “In the balance,” *Nature*, vol. 417, p. 385, 2002.

[19] J. D. Farmer and N. H. Packard, “Evolution, games, and learning: Models for adaptation in machines and nature. An introduction to the proceedings of the CNLS Conference, Los Alamos, May 1985,” *Physica D*, vol. 22, pp. vii–xii, 1986.

[20] J. Ferrell, “Q&A: Systems biology,” *Journal of Biology*, vol. 28, 2009.

[21] R. Gallagher and T. Appenzeller, “Beyond reductionism,” *Science*, vol. 284, no. 5411, p. 79, 1999.

[22] J. P. Gollub and J. S. Langer, “Pattern formation in nonequilibrium physics,” *Reviews of Modern Physics*, vol. 71, no. 2, pp. S396–S403, 1999.

[23] J. D. Halley, F. R. Burden, and D. A. Winkler, “Stem cell decision making and critical-like exploratory networks,” *Stem Cell Research*, vol. 2, no. 3, pp. 165–177, 2009.

[24] J. D. Halley, K. Smith-Miles *et al.*, “Self-organizing circuitry and emergent computation in mouse embryonic stem cells,” *Stem Cell Research*, vol. 8, no. 2, pp. 324–333, 2012.

[25] J. D. Halley and D. A. Winkler, “Critical-like self-organization and natural selection: Two facets of a single evolutionary process?” *BioSystems*, vol. 92, no. 2, pp. 148–158, 2008.

[26] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, “From molecular to modular cell biology,” *Nature*, vol. 402, pp. C47–C52, 1999.

[27] D. Howe, M. Costanzo *et al.*, “Big data: The future of biocuration,” *Nature*, vol. 455, no. 7209, pp. 47–50, 2008.

[28] L. P. Kadanoff, “Chaos: A view of complexity in the physical sciences,” in *From Order to Chaos II Essays: Critical Chaotic and Otherwise.* World Scientific, 1987.

[29] T. Kalkan and A. Smith, “Mapping the route from naive pluripotency to lineage specification,” *Phil. Trans. R. Soc. B*, vol. 369, p. 20130540, 2014.

[30] S. Kauffman, *At Home in the Universe.* Oxford University Press, 1995.

[31] J. Kim, J. Chu, X. Shen, J. Wang, and S. H. Orkin, “An extended transcriptional network for pluripotency of embryonic stem cells,” *Cell*, vol. 132, no. 6, pp. 1049–1061, 2008.

[32] A. D. Lander, “The edges of understanding,” *BMC Biology*, vol. 8, no. 1, p. 40, 2010.

[33] Y.-H. Loh, Q. Wu, J.-L. Chew, V. B. Vega, W. Zhang, X. Chen, G. Bourque, J. George, B. Leong, J. Liu, K.-Y. Wong, K. W. Sung, C. W. H. Lee, X.-D. Zhao, K.-P. Chiu, L. Lipovich, V. A. Kuznetsov, P. Robson, L. W. Stanton, C.-L. Wei, Y. Ruan, B. Lim, and H.-H. Ng, “The Oct4 and Nanog transcription network regulates pluripotency in mouse embryonic stem cells,” *Nature Genetics*, vol. 38, no. 4, pp. 431–440, 2006.

[34] B. D. MacArthur, A. Maayan, and I. R. Lemischka, “Toward stem cell systems biology: From molecules to networks and landscapes,” *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 73, pp. 211–215, 2008.

[35] G. Martello and A. Smith, “The nature of embryonic stem cells,” *Annual Review of Cell and Developmental Biology*, vol. 30, pp. 647–675, 2014.

[36] M. D. Mesarovic, S. N. Sreenath, and J. D. Keene, “Search for organising principles: understanding in systems biology,” *Systems Biology*, vol. 1, no. 1, pp. 19–27, 2004.

[37] S. Muñoz Descalzo, P. Rué *et al.*, “A competitive protein interaction network buffers Oct4-mediated differentiation to promote pluripotency in embryonic stem cells,” *Molecular Systems Biology*, vol. 9, p. 694, 2013.

[38] J. Nichols and A. Smith, “Naive and primed pluripotent states,” *Cell Stem Cell*, vol. 4, no. 6, pp. 487–492, 2009.

[39] G. Nicolis and I. Prigogine, *Self-Organization in Nonequilibrium Systems.* John Wiley & Sons, 1977.

[40] H. Niwa, J.-i. Miyazaki, and A. G. Smith, “Quantitative expression of Oct-3/4 defines differentiation, dedifferentiation or self-renewal of ES cells,” *Nature Genetics*, vol. 24, no. 4, pp. 372–6, 2000.

[41] G. Parisi, “Statistical physics and biology,” *Physics World*, vol. 6, pp. 42–47, 1993.

[42] S. Stepney, “A pattern language for scientific simulations,” in *Proceedings of the 2012 Workshop on Complex Systems Modelling and Simulation, Orleans, France, September 2012*, S. Stepney, P. S. Andrews, and M. Read, Eds. Luniver Press, 2012, pp. 77–103.

[43] S. Stepney and P. S. Andrews, “CoSMoS special issue editorial,” *Natural Computing*, vol. 14, pp. 1–6, 2015.

[44] S. Stepney, P. S. Andrews, and M. Read, Eds., *Proceedings of the 2012 Workshop on Complex Systems Modelling and Simulation, Orleans, France, September 2012.* Luniver Press, 2012.

[45] S. Stepney *et al.*, *Engineering Simulations as Scientific Instruments.* Springer, 2016, [in prep].

[46] K. Struhl, "Gene regulation. a paradigm for precision," *Science*, vol. 293, p. 10541055, 2001.

[47] J. Teles, C. Pina *et al.*, "Transcriptional regulation of lineage commitment – a stochastic model of cell fate decisions," *PLOS Computational Biology*, vol. 9, no. 8, p. e1003197, 2013.

[48] T. Vicsek, "The bigger picture," *Nature*, p. 131, 2002.

[49] D. J. Weatherall, "Phenotype-genotype relationship in monogenic disease: lessons from the Thalassemias," *Nature Reviews Genetics*, vol. 2, pp. 245–255, 2001.

[50] H. Xu, C. Schaniel, I. R. Lemischka, and A. Ma'ayan, "Toward a complete in silico, multi-layered embryonic stem cell regulatory network," *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, vol. 2, no. 6, pp. 708–733, 2010.

[51] P. Zandstra and G. Clarke, "Computational modeling and stem cell engineering," in *Stem Cell Engineering*, R. M. Nerem, J. Loring *et al.*, Eds. Springer, 2014, pp. 65–97.

Action horizon: on the controllability of complex systems, moving towards management for energy systems

Jose Juan Hernandez[1], Jose Evora[1], Enrique Kremers[2], and Johannes Schäuble[3]

[1] SIANI, University of Las Palmas de Gran Canaria, Las Palmas, Spain,
josejuanhernandez@siani.es, jose.evora@siani.es
[2] EIFER, European Institute for Energy Research (KIT & EDF), Karlsruhe, Germany,
enrique.kremers@eifer.org
[3] Chair of energy economics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany,
johannes.schaeuble@kit.edu

Abstract. Building robust and resilient systems is a major challenge for engineering. To this end, analytical indicators may be useful for managers in making decisions. In this paper, a new analytical construct is presented: the action horizon. This construct is devoted to assisting in the control of a system when an event driving the system to its collapse is detected. This construct is useful for analysing the best moment to start executing a corrective action. In this paper, the action horizon is experimentally evaluated in the predator-prey system and we discuss how it could help with managing power systems.

1 Introduction

Instability in power grids can provoke a dramatic power drop, which may lead to service disruption. This situation can happen due to natural environmental changes, machinery failure and/or human actions.

The practical importance of stability analysis, for events that cannot be predicted, is that it helps in selecting countermeasures to avoid system collapse and enhances stability. There are short-term and long-term countermeasures that comprise system planning, operational planning and real time monitoring.

However, once the destabilising event has been identified and the countermeasure has been selected, the following questions arise: when should this action be executed and, more specifically, what is the best moment to execute it in order to minimise costs?

In general, the controllability of a system involves not only the selection of the countermeasure, but also the analysis of the optimal moment to start executing it. However, due to the dynamism of a complex system, it is interesting to delay the execution of the countermeasure since other events could occur and allow the system to recover stability without taking any action. In some cases, therefore, it could be better to do nothing.

This paper introduces the construct of action horizon which helps decide the best moment to execute a countermeasure improving the controllability of complex systems. The structure of the paper is defined as follows: firstly, the controllability concept is presented. After that, this concept is discussed with respect to energy systems and the challenges to improve it. Then, the action horizon is formally defined. This construct is evaluated experimentally in the predator-prey system. The final sections discuss the benefits of the action horizon for the predator-prey and how it could help to manage energy systems.

2 Complex systems and controllability

Complex Systems science has historically emerged from domains such as Sociology and Biology. Complexity theory aims to understand how these usually natural, large and very dynamic systems function, which are their rules and patterns and how global emergence leads to an equilibrium in which the system remains. The concept of Emergence, which describes a global behaviour that cannot fully be inferred by the behaviours of individual agents or, as Aristote long ago observed: when the whole is more than the sum of the parts, is characteristic to all Complex Systems. In an effort to understand Emergent phenomena, different methods of the domain have been used, for example modelling of individual parts of the system by reflecting their behaviour in a synthetic way. This has led to a better understanding of the system; exploring it through detailed inspection and analysis of the individual elements, and reconstructing their interactions and dynamics in order to understand the aggregated behaviour at different levels of this system.

For many years now, however, Complexity Theory has not only helped to improve our knowledge of existing natural systems; it has also has been successfully applied to understanding the behaviour of large man-made complex systems. Evolution theories have been applied to urban systems [2, 3], and the world-wide web, as one of the largest and most widely distributed systems, has been shown to comply to many of the laws that were previously discovered through the exploratory approach. The difference here is that man-made systems are usually designed to fulfill a concrete purpose. Complex systems science as a means for exploring and understanding systems has already lead to scientific breakthroughs (examples of understanding of firefly synchronisation, behaviour in bee hives, etc.).

Going one step further and understanding how we can modify the trajectory of these systems however is an important challenge. [6] raised this topic in their paper entitled "Controllability of complex networks", focusing on finding the entities of the system which have a greater impact on its *controllability*, using the statistical physics approach. Indeed, the methods of exploring complex systems to better know the means by which we can drive those systems into a desired trajectory could be seen as one of the main challenges in this field. Complexity theory could help us better understand engineered systems, such as power grids or other infrastructure networks. However, a transfer of these rather *theoretical* approaches to concrete application cases is needed. In this paper, we initially try to shed light on controllability from a conceptual point of view, then we will try to illustrate its possibilities for typical academic complex system examples, and further apply these methods to a real engineered system. As mentioned above, energy systems are complex man-made systems, nowadays with a tendency towards more decentralised and distributed management patterns. With a view to ensuring both the reliability and sustainability of those systems, a complex systems approach might help to improve their controllability and ensure stability.

3 Energy system controllability

Energy system complexity comes from the great variety of technical installations (e.g. plants, grids), energy sources and carriers (e.g. gas, electricity), actors (e.g. producer, network operator, industrial and private consumer), and is strongly influenced by various economic (e.g. energy prices, market systems), political (e.g. taxes, subsidies) environmental (e.g. weather) and social (e.g. acceptance of technologies) factors. Energy systems nowadays have changed tremendously, in comparison to earlier energy systems. The share of renewable energy generation (REG) in electricity systems is increasing, new actors, for example, traders, have entered the systems, other actors have changed their role (for example, consumers becoming producers), markets systems have changed from a highly regulated to a more liberalized operation, acceptance of certain technologies (e.g. nuclear power) has decreased in certain countries, and information technology has found its way with an extensive application into the systems (see [4]). A wide range of ideas to cope with these new system elements and influences have been developed: inter alia: system integration of REG via smart grids, demand side management (DSM), interconnection of the different energy systems with the electricity system for transport (e.g. via electric vehicles), heat (e.g. via heat pumps), gas (e.g. via power-to-gas), increased use of storage, grid expansion (see [9] for further concepts).

With a growing complexity of energy systems, the corresponding models which aim to analyse and optimize their performance have increased in complexity, too. To keep these models manageable (and, in turn, the real systems they model) they have been simplified, concentrating on the improvement of a subset of objectives ([7]). In most of the cases this simplification merely consists of decreasing the number of system elements and target variables by e.g. limiting the observation period, its differentiation and the geographical extension. Depending on the chosen methodology the complex interactions and procedures normally are mapped as precisely as possible, within the defined subset of targets to study. Therefore, these models represent adequate states for the modelled time horizons and impacts of simulated events and actions. Numerous energy models study a large number of the influences of decisions (or actions, or countermeasures) on the system. Non-deterministic simulations have become an important means of modelling energy systems, as they represented these extremely complex systems very well. Multi agent models (using the same methodology as the model we will investigate later) have become more and more common ([10]). Controllability of energy systems has always been an issue, but with deterministic model actions to control the mode (and thus the represented system), it is relatively easily determined (type and point in time). With multi agent models, controllability becomes a bigger issue.

To illustrate this problem we will present two (of course there are many more!) problems in energy systems: The first problem comes from the increasing penetration of photovoltaics and other small-scale electricity generations in many distribution grids around the world. As such investments have been subsidised for a long time, a lot of electricity consumers have chosen to invest and become producers as well. In fact, during some hours of the day, this generation outstrips consumption, which leads to a feed-in of energy at distribution grid level. The problem here is that most existing distribution grids were not designed for such usage. Because of high security standards, grids in countries like

Germany are heavily oversized in regard to effective power transmission, a fact which has allowed the grids to cope with the new application until now. Other aspects of the electrical grids, such as maintenance of voltages and stability frequency may pose problems, as there are strict regulations of the level for all of them (power quality: $400V \pm 10\%$, frequency: $50Hz \pm 0.2\%$).

New types of big consumers, such as electric vehicles, may also contribute to this problem. Owners of the electric vehicles may decide to recharge them at the same moment at home, which will result in a huge peak load. To deal with these issues, strategies such as DSM in smart grids have been developed. In this approach, flexible demand is shifted using different kinds of incentives, such as electricity prices varying during the day, or a direct control by an operator, for example an electric vehicle fleet operator that schedules vehicle charging. As effective power supply cannot be predicted very well (using weather forecasts, etc.) maintenance of voltage and stability of frequency is more complicated and depends on grid conditions and other short-term factors. Scheduling actions to prevent problems and using these actions optimally (concerning grid conditions and costs) is an issue studied in a lot of energy model simulations.

The second problem arises on a greater scale. As regenerative decentralized electricity generation is increasingly being adopted to confront climate change in many countries, its share in the generation mix increases. This generation is mainly fluctuating and not evenly distributed geographically, due to weather and climate conditions. Therefore, the role of balancing energy, supply and demand, becomes more and more important. To enhance this balancing, energy storage and increasing flexibility of demand may become important. As mentioned before, power grid regulations and safety management are strict. Therefore, in a possible future energy system with a high percentage of uncertain and unequally distributed generation, the question of how much balancing energy has to be stored, and when and where this will be done, is becoming more and more important.

4 Defining the action horizon

An operator in the energy system should be able to analyse the optimal moment to start executing countermeasures. To this end, for controllability purposes, it is interesting to know: (1) the maximum time after an event occurs that the system is still guaranteed to recover its stability by executing the countermeasure; (2) the time after the event in which the action will be effective, with a probability rate; or (3) the time after the event in which the action will no longer be effective.

Understanding these questions evolves around the event horizon concept, which is defined in the general theory of relativity to explain how space time bends around a massive object. This theory predicts that there is a region near black holes from which light is unable to escape. The boundary of this region is the event horizon, or “a point of no return”.

In an analogous manner, if we delay the execution of a countermeasure, there also could be a “point of no return”. That is, a moment in time where this corrective action is not effective for recovering the stability of the system.

This is the proposed **action horizon** construct, defined as the moment up to which it is possible to recover the stability of the system by executing the selected corrective action for a probability of success.

Given a corrective action from a set of possible countermeasures, it is defined a function F_ω that assigns a success probability to the delay of starting the execution of the action.

$$F_\omega : \tau \rightarrow \mu$$

where $\omega \in \Omega$ is a corrective action from a set of countermeasures, $\tau \in [0, \infty)$ is the time from the moment when the event that destabilises the system occurs, and $\mu \in [0, 1]$ is the measure that represents the success probability of a corrective action.

Thus, given a probability μ, it could be calculated τ such that $F_w(\tau) = \mu$. The following properties can be defined for this function:

Non negativity:

$$F_w(\tau) \geq 0, \forall \omega \in \Omega \; and \, \forall \tau \in [0, \infty)$$

Monotonicity:

$$F_\omega(\tau_x) \geq F_\omega(\tau_y), \forall \tau_x, \tau_y \in [0, \infty) \; and \, \tau_x \leq \tau_y$$

Axiom 1. When the system is stable, $F_\omega(\infty) = 1, \forall \omega \in \Omega$, since the action is not required to be executed.

Axiom 2. When the system has collapsed, $F_\omega(0) = 0, \forall \omega \in \Omega$, since the corrective actions will not be able to recover the system at any moment

Then, the **action horizon** for a given action and a defined success probability, θ^μ_ω, is the time after the event occurs such that:

$$\theta_\omega^\mu \in [0, \infty)$$
$$F_\omega(\theta_\omega^\mu) = \mu$$
$$F_\omega(\tau_i) \leq F_\omega(\theta_\omega^\mu), \forall \tau_i \geq \theta_\omega^\mu$$

Since the most common desired success probability is 100%, a simplification of the action horizon notation is:

$$\theta_\omega \equiv \theta_\omega^1$$

5 Experimental evaluation of the action horizon: the predator prey system case

In this section, previously mentioned ideas are applied to a system that has been modelled as an agent-based model: the predator prey system. We have chosen this system as it is simple enough to illustrate the ideas explained in this paper, in addition to being a well-known system.

The model we have used for running all the experiments can be downloaded from here [8]. This model may be freely reused. Our only requirement is to cite us.

To find out the action horizon of the system, we have studied how the system behaves when parameters are stable. Then, an exogenous event causes the system to lose stability, provoking its collapse after some time. An action has been found that, when applied, prevents the collapse of the system. This action, that counteracts the exogenous event, avoids the collapse of the system if applied during the very first step in which the event starts. At this point, an analysis is run in order to calculate the probability of avoiding the collapse of the system when performing this action, depending on the time at which it is applied, after the start of the event. The “action horizon” will be the final time at which the application of the action has a hundred percent chance of success.

5.1 Predator prey model

The model used for analysing the action horizon is inspired by the model developed in [11]. This model considers three types of species: “grass”, “hares” and “foxes”. The predators of the grass are the hares and the predators of the hares are the foxes. This model represents a 2-dimensional toroidal world where every kind of agent is spatially located and the possibility of eating is related to the agents that are in the same cell, as well as their kind.

In this model, there are as many grass units as cells in world. This grass has a specific regrowth time which defines the steps to have new edible grass after the old grass unit has been eaten. This means that every grass unit has an attribute that defines if it is ready to be eaten or not. The regrowth time is the same for all the grass units.

Both animals, hares and foxes, have energy as an attribute and can perform these actions at each step: move, eat and reproduce. The energy is decremented at each step, meaning that an animal may die after having taken several steps without eating (energy = 0). Actions are described in the following list:

- **Move**: animals can only move to adjacent cells and their orientation is calculated randomly.
- **Eating**: as previously mentioned, eating is only possible when there is something to eat in the cell where the animal is. The restrictions are that a hare cannot eat another hare or a fox, nor can a fox eat another fox, or grass. Obviously, when a fox eats a hare, the hare dies and its energy becomes 0. When a hare eats grass, the grass cannot be eaten again until it is regrown.
- **Reproduce**: each kind of animal has a reproduction rate. In each step, a random number is calculated in order to see if the animal has a baby or not. Every animal is able to produce babies, and there is no gender consideration.

5.2 Stability of the system

The predator-prey system remains quite stable whenever all attributes, such as the reproduction rates, the regrowth time of the grass or the initial populations, stay the same along the simulation. In this section, the parameterisation of the system is illustrated in table 1.

Table 1. Parameters of the system

Parameter	**Value**
World size	150 * 150 cells
Grass regrowth time	17 steps
Hares' reproduction rate	4%/step
Foxes' reproduction rate	3%/step
Initial edible grass	50% of all cells
Initial hares	2000 hares
Initial foxes	800 foxes

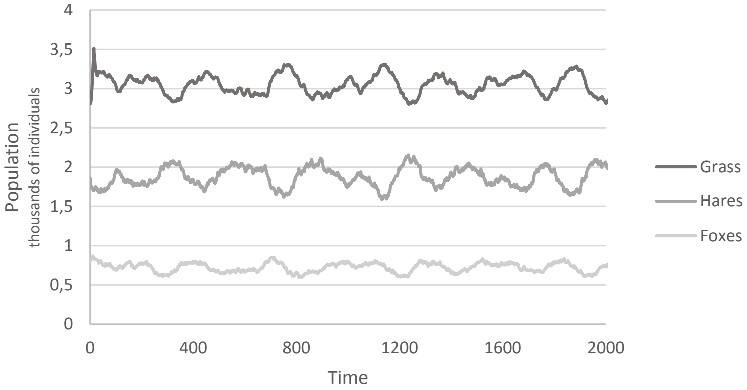


Fig. 1. Population evolution with stable parameters

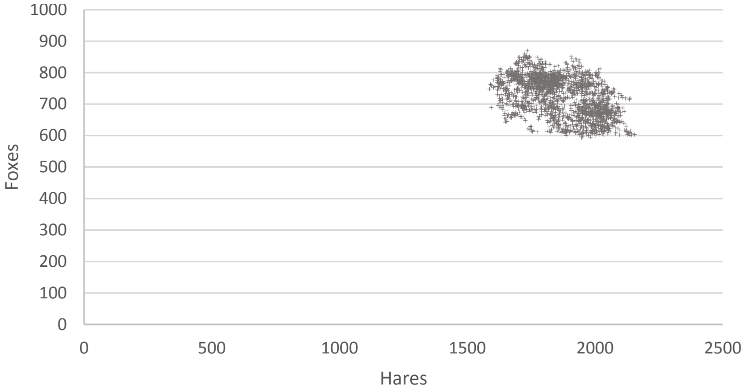


Fig. 2. Relation between hares and foxes populations

As it can be seen in figure 1, the population of the species is stable along the time. Edible grass oscillates between 2,500 and 3,500 units. As it can be seen in figure 2, the fox population oscillates between 600 and 900 and hares between 1,600 and 2,000.

5.3 Destabilising the system

Keeping the system stable results in the action horizon being infinite. That is, if no action is made, the system can remain stable indefinitely. The calculus of the action horizon must be considered whenever some indicators show that the system is moving out from its stable state.

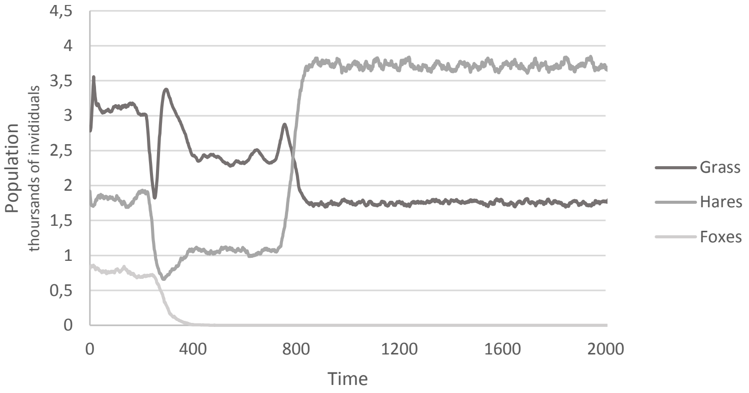


Fig. 3. Population evolution in case of drought

There are many ways to destabilise this predator prey system we are analysing. We have chosen one of those most likely to happen: a drought. The way in which this drought has been implemented in the model is through increasing the regrowth time from 17 to 50. This means that there will not be as many edible grass units as before. Therefore, the hares will be in danger and if their population decreases, the foxes will be in danger too. The duration of this drought is 500 steps.

In figure 3, the evolution of the populations is presented in a simulation in which, after step 200, a drought starts which makes the regrowth time of the grass longer. In this figure, it can be seen how the quantity of grass is dramatically reduced, thus affecting the hare population. This reduction of the hare population provokes, on the one hand, the extinction of the foxes some steps later and, on the other hand, the stability of edible grass at around 2,500 units. When the drought finishes (step 700), a new stability situation is created for both the grass units and the hares. Unfortunately, the overall system has collapsed, as foxes have totally disappeared.

The scattered plot presented in 4, shows the evolution of both the hare and fox populations. It can be seen how, during the drought, the hare population decreases and, as a consequence, the fox population also decreases until they are totally extinct. After, the drought, the hare population increases, reaching around 4,000.

5.4 Stabilising the system

Before analysing the action horizon, an action must be found which avoids the collapse of the system. There are many actions that can be

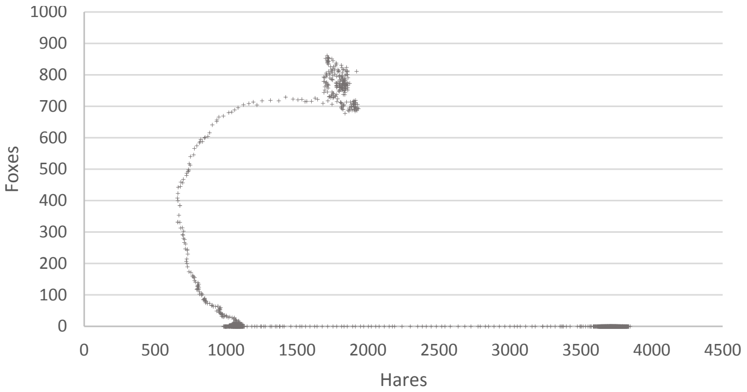


Fig. 4. Relation between hares and foxes populations in case of drought

performed. However, the most ecologically-friendly we have seen is to artificially feed hares by placing some grass in the world, in order to avoid a high decrease in the hare population.

Specifically, the idea consists in leaving some grass (3% of the cells of the world) at every step after the drought event starts. However, this grass is only left whenever the population of hares is lower than 1,800. In figure 5, the evolution of the populations can be observed. There is a high oscillation of the grass population as consequence of the action. Furthermore, the hare population remains stable at around 1,800 whereas the fox population decreases slightly. After the drought, grass, hares and foxes are still alive, so that after some steps the situation before the drought event is recovered. The evolution of both hares and foxes can be observed in the scattered plot presented in figure 6.

5.5 Evaluating the action horizon

After analysing the stability of the system, how to destabilise it and how to find a corrective action, a, to avoid its collapse, the action horizon can be evaluated. The idea of this last part of the experiment is to find the final time at which the previously mentioned action can be performed, so that the probability of the system recovering is 100%.

To run this experiment, a set of simulation has been defined in which the corrective action is applied at different times after the drought event starts. The set of times goes from 0 to 600, with a step of 10. That is, there will be simulations in which the action will start at the step 0, others at 10, 20, ..., 600 after the drought event starts. Every configuration will be simulated 2,000 times so that the probability of success in

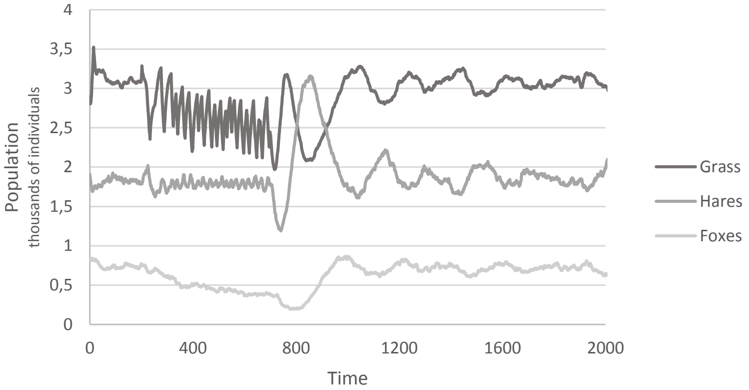


Fig. 5. Population evolution in case of drought and the corrective action is executed

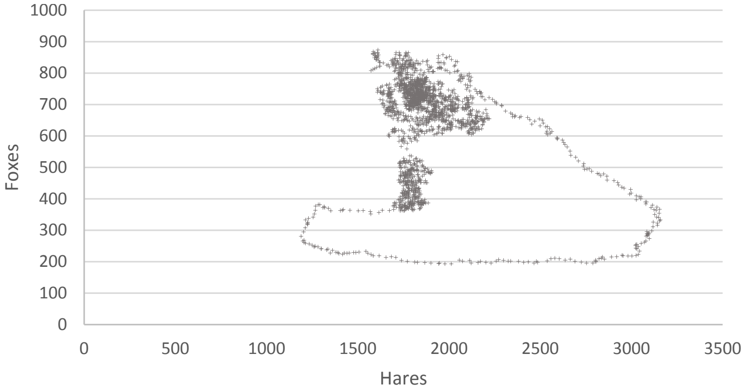


Fig. 6. Relation between hares and foxes populations in case of drought and corrective action executed

applying the action will be calculated for each of those temporal configurations.

In figure 7, the probability of success in applying the action after the drought starts are presented. As can be seen, the action can be applied with a 100% probability of success if applied within 110 steps after the drought event. Then, probability reduces until reaching 500 steps, where the probability of success is close to 0%. Thus, we can determine that the action horizon, θ_a, is 110 steps after the drought event considering action, a.

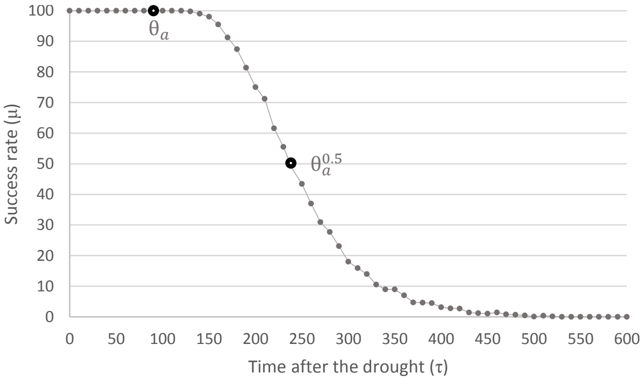


Fig. 7. Probabilities of succeeding in applying the action after the drought starts $F_a(\tau_i)$

6 Discussion

In conducting these experiments, we have relied on three main factors that seem to be the most critical for the action horizon:

- **Event**: the event that produces the systems collapse determines how far away the action horizon is. If we had set the regrowth time to infinite, instead of 50, the action horizon would have definitely been quite shorter. Moreover, the action that we applied to correct the situation (putting grass in the 3% of the cells) may not have been sufficient to recover the system.
- **Action**: the action to avoid the collapse of the system is also critical to determine how far away the action horizon is. In the case we have presented, the action horizon would have been further away if the grass had been set in more places, or the condition to leave grass had been when hares population was lower than 2,000. Obviously, if the action is too aggressive, it is possible to bring the action horizon closer, as a collapse may be caused by the action itself.
- **State before the event**: this last factor is not as critical as the other ones, but can determine how far away the action horizon is. If the system is running with low populations at the time in which the event happens, the action horizon will be closer, as an action to recover the system will be required before.

Another important factor to consider when evaluating the action horizon is the systems stochasticity. Under the same conditions in the three factors previously mentioned, the simulations can reach different action horizons as the evolution of the populations is determined by stochastic events.

Knowing the action horizon allows the managers to decide when is the best moment to act in the system so that effort can be reduced. In this case, one of the low-cost solutions might be to act within the action horizon time, as this means that effort was saved in all the previous steps in which nothing was done. Nevertheless, it is important to know the probability of succeeding after the action horizon, since this may be helpful in deciding what kind of action to apply, once this temporal point is reached. That is, if it is known that an action has a certain probability of success, perhaps it will be possible to come up with a better action that may have a higher probability of success.

7 Conclusion

In this paper we focused on a concrete example based on the predator-prey model in order to show the action horizon concept and how it supports the controllability of a complex system. By simulating a destabilising event, the system was disrupted, making it unstable. At this point, corrective actions were performed and tested in order to assess the so called action horizon. This is defined as the maximum time after the event occurs in which a corrective action has a desired success rate in achieving stability.

It is interesting to know the action horizon in order to delay the execution of the countermeasure. Due to the dynamism of a complex system, other events could occur and allow the system to recover stability without executing any action.

Another factor that should be considered by managers is the cost of executing the actions. The goal of minimising operational costs requires the analysis of the best moment to act in order to recover the system.

For example, in the case presented, it could be wise to wait for the action horizon with a 100% rate of success. In this way, new events can be seen that may restore the system stability, so that no intervention is needed (e.g. drought finishes), or other events that may push the system to other kinds of risks, in which case this corrective action could be harmful.

Future research on action horizon could analyse operational costs. In this way, it could be calculated what is the action horizon of the combination of two different actions or how several events occurring at the same time influence the action horizon. Further studies should also analyse the costs of recovering the system from its collapse, since this could be less costly than avoiding its collapse. For instance, feeding hares may be more costly than bringing foxes in from other territories after the drought has ended.

Furthermore, additional events could be considered in order to see how it could affect to the Action Horizon. Considering the combinatorial that this may have, in order to reduce the computational costs, it would be interesting to apply techniques for calculating the number of runs required to have a certain level of certainty when defining the Action Horizon [1].

8 Outlook

In the balancing of energy systems, the risk of blackouts has been thoroughly analysed. For example, frequency drops due to power unit failures may lead to blackouts and even complete desynchronisation of the system. In [5] the impact of distributed load management on system frequency was analysed. Furthermore, controllability of the energy system in terms of voltage control could be analysed. By optimising the use of flexible loads at the right moments and right places, voltage drops can be reduced. The analysis of the action horizon for these cases would lead to a measure that could serve as a stability indicator of the grid, as well as system frequency. Taking the flexibility potential into account, this indicator could show the added value of using local load management, and serve as risk assessment in real time for grid operation.

The concepts and methodology will be modelled within an energy system model that allows evaluating scalable, distributed agent-based load management strategies. The energy system will be a smart grid at distribution grid level: the modelled distribution network has certain flexibility potential. Therefore an improved organization and utilization of its elements will be performed, using a combined methodical approach based on agent-based simulation (ABS) and energy system model (ESM) optimization. As the optimization is deterministic, it will not take into account action horizons, but as the agents performing the optimization are placed in an ABS, the environment and its agents will be able to perform actions that have an action horizon. Flexible load (and generation) can be supplied by different technologies, such as some already existing appliances, demand side management (DSM), new system elements, for example, electric vehicles (EV), stationary batteries, or decentralized generation (for example with photovoltaic panels or microCHP). In order to ensure an efficient operation of the electricity system, these examples of flexibility can support the central generation system in dispatching and grid balancing which represents the previously mentioned countermeasures (or actions).

Conventional optimization of the integral system becomes very complex as the degrees of freedom in a system with large amounts of distributed resources normally is very high. Therefore, modular and scalable management techniques are required to tackle large amounts of distributed resources. For this reason, in our forthcoming work heuristics based on agent-based models simulation, including the concept of this paper and optimization of system parts will be analysed and tested to explore a scalable and multi-level solution in the given context.

The generic approach will be applied to a use case for which we will model a differentiated load modelling at household level (called micro-modelling, for detailed representation of load shedding and demand-side management). Further on, the aggregation of loads to low-voltage transformer and the development of distributed, local demand side management mechanisms, based on local optimisation including technical and economical perspectives will be performed. Local and global constraints will be taken into account and the iteration with a grid model will allow identifying local grid constraints on the distribution network.

References

[1] K. Alden, M. Read, J. Timmis, P. S. Andrews, H. Veiga-Fernandes, and M. Coles, "Spartan: a comprehensive tool for understanding uncertainty in simulations of biological systems," *PLoS Comput Biol*, vol. 9, no. 2, p. e1002916, 2013.

[2] M. Batty, *Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals.* The MIT press, 2007.

[3] M. Fujita, P. Krugman, and T. Mori, "On the evolution of hierarchical urban systems1," *European Economic Review*, vol. 43, no. 2, pp. 209–251, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001429219800066X

[4] IEA, *World Energy Outlook*, 2014.

[5] E. Kremers, J. M. Gonzlez de Durana, and O. Barambones, "Emergent synchronisation properties of a refrigerator demand side management system," *Applied Energy*, 2012, http://dx.doi.org/10.1016/j.apenergy.2012.07.021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261912005338

[6] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabasi, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011, 10.1038/nature10011. [Online]. Available: http://dx.doi.org/10.1038/nature10011

[7] J. Schäuble, P. Jochem, and W. Fichtner, "Impacts of electricity consumers unit commitment on low voltage networks," *Operations Research Proceedings 2014, Selected Papers, forthcoming*, 2015.

[8] Siani, "Action Horizon Evaluation on predator prey system," 2015. [Online]. Available: https://github.com/sianiresearch/predatorprey

[9] G. Verbong and F. Geels, “The ongoing energy transition: lessons from a socio-technical, multi-level analysis of the dutch electricity system (19602004),” *Energy Policy*, vol. 35, no. 2, pp. 1025–37, 2007.

[10] A. Weidlich and D. Veit, “A critical survey of agent-based wholesale electricity market models,” *Energy Economics*, vol. 30, no. 4, pp. 1728–1759, 2008.

[11] U. Wilensky, “NetLogo Wolf Sheep Predation model,” 1997. [Online]. Available: http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation

CoSMoS in the Interactive Simulation Curriculum

Sebastian von Mammen, Sarah Edenhofer, and Jörg Hähner

University of Augsburg, Augsburg, Germany

Abstract. Animations at first, then real-time computer graphics and human-computer interaction techniques have made interactive simulations possible. Nowadays, they play an enormously important role in training the operation of complex technology such as aircraft, and they have achieved a remarkable share in the computer gaming industry. The fast emergence of virtual and augmented reality solutions promises an even wider spread and a greater impact for interactive simulations in the near future. Due to the multifaceted nature of interactive simulations in terms of confluent scientific fields, due to the underlying iterative and agile development processes, and last but not least due to the inherently central human factors, we have been integrating the CoSMoS process of complex system modelling and simulation into our course curriculum on interactive simulation for computer science graduate students. In this work, based on an overview of the contents and the logistics of the course, we present our conceptual efforts towards this goal. We emphasise the role of the CoSMoS process, discuss its impact on the students' projects, and we provide concrete examples.

1 Introduction

Since the first human-in-the-loop simulators entered the market in the 1980s [44], interactivity has evolved into an increasingly important aspect of scientific simulations. Nowadays, established mathematics frameworks such as Mathematica, Maple or Matlab provide ample support for visualisation routines and interactive parametric exploration of any devised models, whereas development frameworks such as Unity3D, Unreal Engine or CryEngine that primarily target the computer gaming market are offered and marketed in the context of simulations as well. Due to the fast-paced strides towards ubiquity of virtual and augmented reality systems [59], for instance by utilising widely availably smart phones, we expect an even more accelerated spread of interactive simulations in the near future.

Numerous areas of computer science feed into the development of an interactive simulation—human-computer interaction, real-time computer graphics, visualisation, modelling and simulation approaches, etc. The according methodologies and techniques are deployed to make a simulation model accessible to the user. In addition to translating reality into an adequate domain model and further into a suitable computational representation, or platform model, the creators of an interactive simulations are confronted by an abundance of user-related interfacing challenges. In all brevity, they need to translate the user's wishes into effective commands of control and model changes, and they need to translate the matter-of-fact results of the simulation process into visualisations (mostly), that are quickly understood and capture rather than lose the user's attention. To render the trade even more challenging, all of these translations need to happen at rather high rates that provide for an uninterrupted interaction experience.

Motivated by their great and growing importance, we set out to teaching students foundational knowledge about interactive simulations. In particular, we designed a university course to empower computer science graduate students with an interwoven in-depth apprehension of methods in the associated fields. Thus, the students acquire knowledge to evaluate and skills to contribute to the design and the programmatic implementation of interactive simulations. In this work, we present our course concept, focusing on the role of the CoSMoS process of complex system modelling and simulation. Based on a description of our course concept (Section 2), we highlight the role of the CoSMoS process in the curriculum as a whole, and with respect to the accompanying student projects, in particular, in Section 3. Next, we present several select student projects (Section 4), also shedding light on the development processes the student went through throughout the term. We conclude this work with a summary of our findings and an outlook on future work on CoSMoS for interactive simulations.

2 Synopsis of an Interactive Simulation Course

An interplay of a variety of computer science disciplines provides the foundation for interactive simulation. Accordingly, the contents of a course on interactive simulation greatly vary dependent on the expected knowledge base of the students as well as complementary courses offered by the hosting institution. In our case, we devised a university course suitable for master students in computer science and closely related programmes of study. The course runs for four months, staging a 2-hours-lecture and a 2-hours-tutorial each week. In combination with the allotted project work, the course demands for a total workload of 150 hours.

In the following paragraphs, we summarise the contents of nine provided lecture units. After an introduction to the subject matter, we teach the CoSMoS approach to modelling and simulation. Next, foundations of computer graphics are conveyed, as well as a mathematical display of real-time physics computation models and algorithms. Visualisation methods and an introduction to human-computer interaction techniques complete the first block of basic lecture units.

The second block of advanced lecture units focusses on model representation and process optimisation both of which are important constituents of interactive simulation technology. After presenting the foundations of discrete event simulation and an array of computational representations, popular conservative and approximative acceleration mechanisms in the realm of interactive applications are discussed. As the versatility and the transferability of an agent-based modelling (ABM) approach is rather unique but can easily result in costly computations, we commit another lecture unit to introducing novel research concepts that promise to scale ABM to reach interactive performances.

2.1 A Short History of Human-in-the-Loop Systems

The history of interactive simulation begins with efforts to enhance existing simulation data by means of interactive custom animations. We present an according example, a SIMAN job shop simulation model of an automatic guided vehicle system visualised by the CINEMA animation system [47]. The optimisation of industrial workflows was the most compelling argument for such animation systems in the 1980s. At the time, the market offered an array of simulation animation tools, including Model Master, XCell, and Performance Analysis Workstation. Next, solutions were offered that tightly coupled interactive visualisation with the underlying simulation. See-Why was one of these packages that promised Visual Interaction Simulation (VIS). An according example allowing the configuration of a locomotive servicing centre is shown [45]. Definitions of basic terms such as model [67], simulation [4], and the early-conceived notion of interactive simulation (‘on-line simulation’) [31] follow the introductory historic examples. The distinctive feature of interactive simulations is the possibility of human influence during the simulation process, typically referred to Human-in-the-Loop systems [52]. We look at the taxonomy of interactive simulations, their advantages over stand-alone simulations, established fields of application, technological challenges, and their historic evolution in respect to programming paradigms, languages, and interfaces. The basic steps taken in a simulation project, especially under consideration of interactivity, and several examples of state-of-the-art interactive simulation systems round off this lecture unit. The examples are organised to guide the students from comprehensive immersive solutions with special hardware configurations (driving and flight simulators) to software-only solutions, which are the focus of the lecture.

2.2 The CoSMoS Process & Gamification

The orthogonal relationship between descriptive and defining models precedes the remaining contents of this lecture that primarily aims at the process of modelling and simulating complex systems. Examples for seemingly disparate approaches are provided that put different weights on these respective modelling purposes: In detail, these are understanding complex behaviours of real-world systems, simulating complex system themes, engineering complex algorithms, and engineering complex systems [49]. We consider the means of scientific instrumentation (extrapolation, conversion, augmentation) to become aware of its limits and limitations [25] and to define the products of the CoSMoS modelling and simulation cycle. The definition of these products motivates an elaborate discussion of the phases of discovery, development and exploration [2]. Interactive simulations need to engage their users. The relatively novel paradigm of turning burdensome chores into games suits this challenge well. Hence, we proceed with the presentation of game definitions and, more specifically, aspects of development of computer games [40]. The short history of serious games (starting in the early 2000s) is summarised [21, 57] and representative examples are demonstrated (e.g. [9]). Their concrete successes in terms of engagement are discussed and a comprehensive list of game design elements [11] is presented that can be utilised to 'gamify' an interactive simulation. All of these elements can be derived from the cornerstones of intrinsic motivation, namely relatedness, competence, and autonomy which are explained as well [33]. In the context of interactive simulations, these aspects can be considered during the discovery phase, whereas gamification typically takes place during the development phase of the CoSMoS process.

2.3 Computer Graphics Foundations

The increasing availability of dedicated graphic processing units (GPUs) promotes the utilisation of a standardised 3D rendering pipeline for any kind of visualisation needs, whether 2D or 3D, vector-based, or otherwise. Therefore, this lecture units seeks to empower the students with a basic understanding of this rendering pipeline [1, 40, 63]. Basic concepts that are presented in this lecture unit are: object definitions based on geometric primitives and various kinds of textures, the view reference, spatial transformations (also introducing quaternions in the context of rotational operations), basic types of lighting, shading, and light sources, shadow definitions and different implementation techniques such as shadow maps. A short walkthrough of generating 3D graphic assets suited for real-time rendering rounds off this lecture unit.

2.4 Real-time Physics

This lecture seeks to provide a solid grip on real-time capable approaches to simulate physical processes [8, 24]. We have a quick look at the taxonomy of the vast field of physics simulation [14] but we focus on real-time methods of forward dynamics, covering three categories: rigid-body dynamics [7, 15], soft-body dynamics [13], and particle physics [38]. Next to the general laws of motion, we look at non-penetration constraints, collision resolution and friction forces, and complementary constraints in the context of rigid body simulation. For calculating the respective forces, we present the penalty force method, Lagrange multipliers, impulse-based simulation, and reduced coordinate formulation as well as the Coulomb friction model. We introduce a taxonomy of constraints and explain how they can serve as representation of mechanical joints of articulated bodies [6]. We follow the steps to transform the resulting differential algebraic inequalities into an efficiently solvable linear complementary problem. We conclude the integration of forces with a brief recap of basic methods of numeric integration, starting with Euler and Runge-Kutta. We discuss algorithms for efficient collision detection and contact point generation, e.g. [18, 37]. We then widen the scope of this lecture unit, looking at one specific approach to computing incompressible deformable mesh dynamics that is superior to alternative approaches in terms of efficiency and accuracy [13]. Finally, we introduce to real-time particle physics, explaining particle approximation functions based on the notion of kernel functions [38], culminating in recent advancements in unified real-time physics simulation [39].

2.5 Visualisation Methods

To a large extent, interactive simulations imply some kind of visualisation of the underlying models and the emerging simulation processes. In this lecture unit, we emphasise the necessity to consider human perception and information processing when crafting the platform model of an interactive simulation and we provide an overview of foundational visualisation techniques. We follow the structure of numerous textbooks on this subject matter and motivate the discussion on the human vision apparatus by providing several examples of optical illusions [62]. In a 7-step guideline, we establish an idea of the selection of the proper visualisation method embedded in the context of data acquisition and the intended modes of interactions [17]. Qualitatively, visuals can be measured in terms of novelty, informativeness, efficiency and aesthetics [56]. We shed light on various scientific, multidimensional, multivariate visualisation methods [65], before we turn to visualisation techniques that allow for the immersive augmentation of simulation contents, such as examples of flow visualisation [41], graph-based visualisations [34], or the transformation of volumetric (4D) data into 3D surfaces [30].

2.6 Human-Computer Interaction Techniques

The design of interactive simulations necessitates an interface between human and computer. This lecture unit provides the necessary background, starting with a brief history of HCI research [42]. Human interaction requires the user to process and translate sensed information into motor activity [12]. In general, user interactions can be classified as operations of selection, manipulation, navigation, and system control [40]. As any other design task, the design of interactions is the result of a tradeoff between multiple goals and constraints [16]. We present a top-down approach to designing interaction scenarios that starts with the definition of an application's requirements and arrives at individual interaction tasks. We have a brief look at multimodal approaches, e.g. affective, perpetual, attentive, and enactive interfaces [29], and we explore the modes of interaction of an embedded multimodal prototype game [19]. We quickly step through established and emerging immersive hardware technologies, including devices of motion sensing and object tracking capabilities. We convey a general understanding for the hard latency limitations of interaction hardware and we provide recipes for rather general issues that arise in real-world sampling, i.e. noisy sensing and the state estimation problem [40, 63].

2.7 Discrete-Event Simulation

In this lecture unit we provide an overview of computational representations as well as modelling and simulation approaches. We start out with explaining the basic terminology of discrete-even stimulation (DES) in the context of previous lecture units, especially those described in Sections 2.1 and 2.4 [3]. To this end, hybrid simulation and combined simulation concepts are of great importance. We roughly trace the history of this seminal field in terms of DES software packages and languages [43]. Three ‘world views’ on DES (event scheduling, activity scanning, and process interaction) serve as the starting point for our venture into historic approaches. We meticulously describe the elements of a DES and provide a glimpse at charts already used for engineering DES back in the 1960s. These diagrams (activity cycles, wheel chart, flow charts) are our point of departure towards other computational representations commonly used for modelling and simulation: Finite state machines, UML transition diagrams, Petri nets [48], artificial chemistries [5], cellular potts [28], cellular automata [68], random boolean networks [32], boids [51], L-Systems [50], swarm grammars [60], and the general approach of agent-based modelling [10, 66].

2.8 Acceleration Algorithms

Low latency requirements of the interaction interface (Section 2.6) as well as the desire to serve large, complex models for interactive exploration at real-time demand for the utilisation of sophisticated acceleration algorithms. Computer graphics and real-time physics are currently occupying this niche and this lecture unit aims at exhibiting their commonly used, highly efficient approaches [1].

It is divided into four parts: First, we focus on bounded volume hierarchies (BVHs) and binary space partitioning trees (BSPs). While BVHs are built bottom-up based on bounding volumes that enclose geometries or other bounding volumes recursively, BSPs are generated top-down by recursive division of the simulation space. We also provide guidelines to coping with (a) mobile and (b) deformable objects [35, 36, 58]. Second, we explain different culling techniques which ensure that graphical objects are not pushed through the rendering pipeline (Section 2.3), if their contributions to the final rendering are marginal or not existing. Examples are surfaces that lie outside of the view frustum, those that are hidden behind objects, or those that are so far away from the camera that they could hardly be seen on the screen. Third, we present approaches that lower the level of detail (LOD) of the rendered objects to the match the actual needs—as opposed to always rendering at the highest possible level of detail [23]. An example of LOD is the number of triangles of discrete geometries which can, for instance, be selected according to the distance to the camera. We conclude this lecture unit showing stochastic acceleration algorithms for collision detection.

2.9 Dynamic Model Abstraction

Motivated by the outlook on large, multi-scale, multi-representation simulations [26], we tackle the issue of ever-growing computational complexity by means of dynamic model abstraction techniques in this lecture unit. Particularly, we focus on adaptive optimisation of agent-based models, as they can serve as a generic computational representation. Concerning the immense computational costs running large-scale simulations, we discuss the limitation of different model aspects and how they could improve efficiency. We conclude this investigation with the realisation that if we want to model and compute natural systems, we need to consider dynamic systems with dynamic interaction topologies (DS^2) [20]. In addition to hardware-based solutions (e.g. [27]), we promote dynamic model adaptation. Agent compression identifies and subsumes clusters of similar agents [55]. The dynamic extension of this approach considers container agents to maintain similar agents and to offer the possibility to remove or add individuals on demand. Compression managers are responsible for (a) organising the container agents and their contents, and (b) representing the compressed agents to the remainder of the model [64]. Taking this idea even one step further, we provide the detailed steps of the self-organised middle-out abstraction approach [61] and we show its capabilities with respect to a decentralised, agent-based blood-coagulation simulation [53, 54].

3 CoSMoS' Central Role

The CoSMoS process is introduced right after a general introduction to the course (see Section 2.2), as it provides a flexible, yet focussed guideline for all phases of the development of interactive simulations. In this section, we first detail a way of applying the CoSMoS process to student projects, following the explanations in [2]. Second, we present a course infrastructure to realise this approach.

3.1 CoSMoS for Interactive Simulation

We discuss the three phases of the CoSMoS cycle (discovery, development, exploration) in the context of interactive simulation based on the five activities performed during each phase: scoping, modelling, experimenting, documenting, and interacting.

During the *discovery phase*, the greatest challenge to the students is the primary need to settle on an application domain and to define the goals of the interactive simulation, e.g. teaching contents or providing for a scientific exploration tool. Although the students appreciate the opportunity to freely chose an application domain for their projects, they seem to be more comfortable when provided with a theme, for instance biology. The only constraints regarding their choice is the projects' evaluation based on the following aspects, which are set to ensure their usefulness and the comparability.

Science The model that drives the resultant prototype has to be scientific, i.e. it has to be based on scientifically published results. A CoSMoS compliant development process certainly supports this endeavour. In addition, the modelling domain, the validity of the modelled system, its degree of innovation, and the computational representation and algorithms used give strong indications for a scientific approach.

Gamification The prototype has to motivate the user to interact and explore the simulation space. One can try capturing this aspect quantitatively by describing interaction possibilities, user guidance, usage of game elements, and the factors of intrinsic motivation as referenced in Section 2.2.

Complexity Interacting with the prototype should be rewarding in itself, i.e. it should convey insights with respect to the underlying scientific model. The model complexity defines the scope of potentially educational contents, given the conveyed complexity considers the full extent of the underlying model.

Aesthetics An interactive simulation has to be aesthetic, not only to efficiently convey information to the user but also to motivate their involvement. Aesthetics can be promoted following established design principles, by utilising beautiful visual assets, and by combining them in novel ways.

Any steps towards desirable domain attributes, concrete domains, and even concrete goals and an application concept, necessitate answering questions about the projects' criticality, their limitations, and their measurable success. In the context of interactive simulations, the answers typically stress the relationship between the software and the user. The utility for the user, for instance, not only considers a final simulation result but also the benefit of pro-active participation in the simulation process. Accordingly, limitations are not only considered regarding the accuracy and efficiency of the simulation but especially with respect to the degrees of freedom exploitable by the user and the quality of the communication between the user and the simulation, including aspects such as clarity and attractiveness. The modelling activity during the discovery phase is rather limited in the scope of a term-long project. Despite the abundance of scientific data accessible through online libraries and the large repositories of computational libraries and tools for numerous scientific domains, comprehending the elements and their relationships of a previously unstudied field is a rather difficult task. For this reason, and also to provide the necessary degree of autonomy to intrinsically motivate the students, we allow the students to decide on a concrete domain and goal by themselves; based on supervisory feedback on a written proposal and classroom presentations with subsequent discussions, the core ideas can then be quickly translated into first proof-of-concept prototypes. The discovery phase is decisively shaped by documentation activity—from coarse to fine grained searches for references and tools, through merging sources, assumptions and ideas into a concept proposal that includes an early domain model, to creating a first prototype that provides evidence for the created line of argument.

During the *development phase*, documentation about the students' activities is similarly important. However, to a great extent, it coincides with the development of the platform model, an accompanying commented code base, and its transcription for a given simulation platform. To help reduce the burden that a comprehensive interactive simulation project incurs, we diminished the scoping activity of the development phase and taught about various tools of the trade for interactive simulation development—ranging from 3D asset creation over scripting and high-level, component based model compositions to utilising third-party plugins and libraries for the targeted development environments. In frequent presentation and feedback sessions, we ensured that domain elements were properly represented and domain behaviours were not directly encoded in the models. Adding instrumentation to the platform model plays an important role for interactive simulations. This step should closely follow the interaction concept developed as an extension of the usual domain model, i.e. one that encompasses the user as a special model element. Nevertheless, the targeted simulation platform may provide a rather special interaction infrastructure. For example, the ubiquity of mobile, multi-touch platforms equipped with relatively weak processing capabilities competes with the processing power, storage capacity, and extensibility of desktop systems. Clearly, any specific interaction platform demands for individual adjustments of the platform model to realise both the interaction and the simulation concept. Experimentation in the development phase begins with the first prototype supporting preliminary user interactions. At later stages of the development phase, it increasingly involves feedback from testers not directly involved in the development work. Beyond honing the visualisation, consistent design, usability and the scalability of their platform models in terms of parameter settings, numbers of interacting agents, increasing levels of difficulty and the fine line between balanced, rewarding interaction and user boredom and frustration.

During the final stage of the course project, the *experimentation phase*, the students focus on logging and analysing user responses to their simulations. To keep the amount of work at a level reasonable in the context of our course, the students are asked to try each others' simulations and to ask their friends and relatives to provide them with some preliminary feedback. This exposure typically already provides comprehensive insights into the users' general interest in the topic, their opinion about model complexity and aesthetics, and whether they think it is educational. Based on these evaluations, the students are encouraged to hone their software and to launch more comprehensive online surveys. However, these more rigorous steps are not mandatory course stipulations. Nevertheless, the gathered preliminary data in combination with the initial motivation of their projects, the development processes and the implementation results, serves as an extensive basis for fleshing out their final report. It culminates in conceptual improvement that could instigate the next development cycle.

3.2 Course Project Infrastructure

Above, we already touched upon the students' deliverables and how their realisation is backed by the CoSMoS process. Now, we briefly present the logistical infrastructure of the course setup to support the traversal of the CoSMoS process throughout the term.

During the first lecture, the students are first informed about the course contents and its stipulations. For the remainder of the lecture, we present and explain several examples of possible project concepts. Although the students may conceive a project idea completely on their own, providing examples proved important to communicate the expected scope and the imparted opportunities. Within ten days' time, teams of two students need to author a proposal of their projects. On two pages (ACM double-column format), the students need to motivate, present and detail their concepts. Hereby, the envisioned user experience plays an important role as it ties different aspects of the envisioned simulation together and it implicitly underlines its goal. From a CoSMoS perspective, the project proposal is part of the documentation activity of the discovery phase. As such it serves not only as a platform for the students to substantiate their initial ideas and consistently brush up their findings but also to communicate their concept to the instructors.

At the time of the proposal submission, a second lecture unit has introduced the general topic of the course (Section 2.1) and a first tutorial session has familiarised the students with the development environment that we recommend (in previous years, we recommended Unity3D). The day after the submission of the proposals, the students are asked to present their concepts in short 3-minute presentations during the tutorial session. In this way, all the students in the course would gain an overview of their peers' projects and learn about new ideas, possibly even about the usage of previously unknown code snippets, etc. The quick start into the projects and presentations early in the term help the students build up momentum for their projects. In fact, until the last few weeks of the term, the students would present the state of their projects bi-weekly. This fosters a certain sense of togetherness and it ensures guidance to maintain high productivity and to avoid frustration.

Two weeks before the end of the term, final reports are due (six pages, ACM double-column format) that should ideally condense the documentation recorded throughout the whole term. One week later, the students need to submit their projects, including batches of slides for the final presentations which are given in front of faculty and students of the whole department. The audience is asked to vote for the best entry in terms of the generic project criteria: science, complexity, gamification, and aesthetics (Section 3.1). A 15-minute brief oral exam at the end of the term makes sure that the students have learned and understood the diverse contents of the course and their relationships.

4 Select Student Projects

In this section, we present select student projects that were developed in two iterations of our interactive simulation course. First, we describe some of the outcomes exemplarily. Second, we shed light on the CoSMoS-driven development process of a specific project.

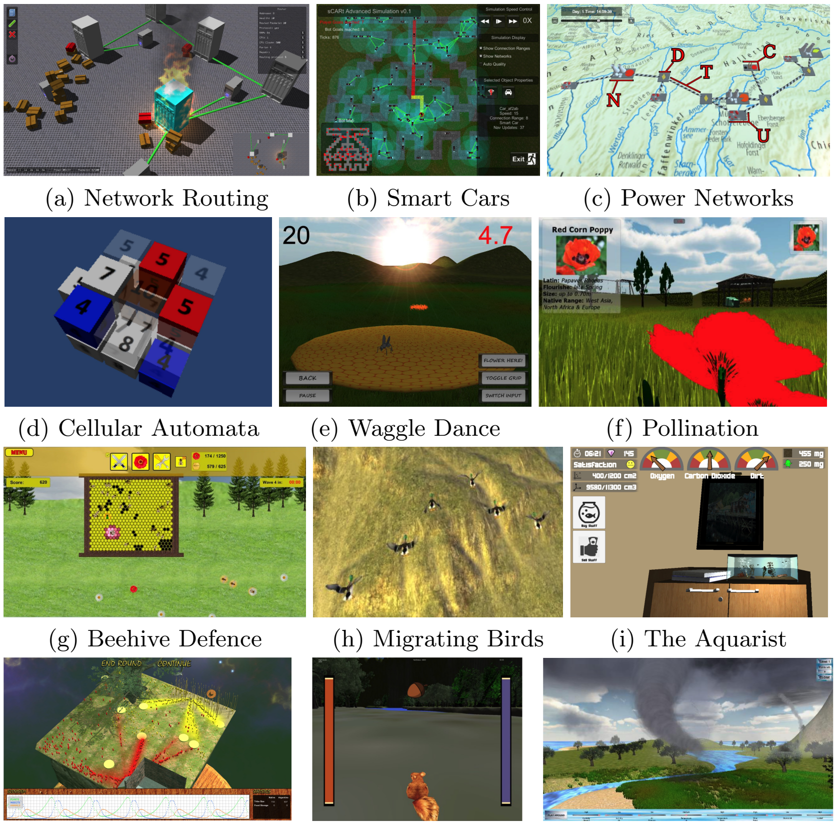
4.1 Examples

During the first iteration of the course, the majority of the students chose "technical systems" topics such as routing in communication networks, smart cars, and power networks. Figure 1(a)-(c) shows according screenshots. The user is tasked to build and maintain power or communication infrastructures to ensure their proper functionality. In the network routing and the smart car example, the user also had to guide the network activity itself by laying out flow paths of the respective traffic. Some students also journeyed towards biological themes such as cellular automata as seen in Figure 1(d). Here, a game of life variant served as the basis for a two-person game with the goal of conquering as much space as possible solely by adjusting the cells' rules. During the second iteration of the course, we proactively advertised biological and natural phenomena as an exciting and multifaceted field to motivate the student projects—yet, they were still free to take their projects into other directions. As a result, three groups let their projects revolve around bees (we had not motivated this trend), see Figure 1(e)-(g). In the first one, the user had to guide a bee's waggle dance to point its peers to the location of a food source outside the hive. Figure 1(f) shows a screenshot of a bee simulation that focusses on the challenge of gathering nectar and thereby helping flowers pollinate. Lastly, a complex real-time strategy simulation is presented in which bees need to gather resources, maintain their hive and defend it against wasp intruders. Other examples included the user-guided migration of a flock of geese (Figure 1(h)) or the establishment of a fine balance of interdependent inhabitants in a simulated aquarium (Figure 1(i)). The interdependency of species provided the basis of yet another title where a new ant species threatens to overrun a native species and the user is tasked to maintain a balance by building barriers or proactively diminishing one or the other ant population (Figure 1(j)). Focussing on solitary species, a squirrel simulator offered the experience of sharing a rodent's worries: collecting, burying, and finding enough nuts to survive the winter season (Figure 1(k)). The importance of climate also inspired "Cloud Computing", where a user was tasked to set the environmental conditions in such a way that certain weather phenomena such as rain or tornados would emerge (Figure 1(l)).

The set of presented examples emphasises the flexibility of the course project in terms of contents, perspectives and goals of the student project while addressing the project requirements as outlined above (Section 3.1). Next, we dive into one specific project and shed light on how the CoSMoS process informed its development.

4.2 A CoSMic Case Study: "Drink & Drive"

One student team decided on creating a serious game about the negative effects of alcohol on traffic participants. They understood that although some accurate simulators exist for this purpose (e.g. [22]), they don't provide for a stimulating, engaging experience. At first the students were hesitant whether their idea was acceptable as it attempted to approach a serious topic in an engaging, fun way. We encouraged them to try anyway. Findings about games that had been developed for this purpose, such as [46], further boosted the students' ambitions. These preceding titles had disconnected from the actual problem too much, for instance by assuming a third-person perspective on the driving situation. Quickly, the students realised that their interactive simulation should fill this gap and make their title "Drink & Drive" both fun and educational, so that



(a) Network Routing (b) Smart Cars (c) Power Networks

(d) Cellular Automata (e) Waggle Dance (f) Pollination

(g) Beehive Defence (h) Migrating Birds (i) The Aquarist

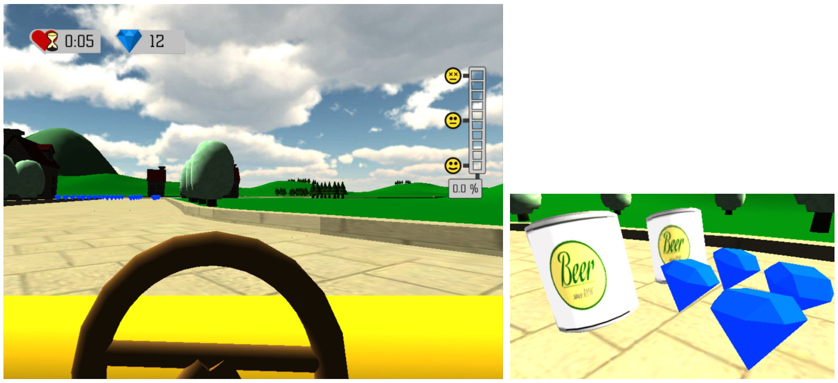
(j) Invasive Species (k) Squirrel Simulator (l) “Cloud Computing”

Fig. 1. Screenshots of interactive simulations developed as student projects in two iterations of the course.

the target group of soon-to-be drivers and young drivers would engage in and learn about this fundamentally serious topic. The second part of the discovery phase of their project shed light on actual models of impairment of drunk drivers. Its last part posed the greatest challenge: Merging the seemingly conflicting concepts of learning about the severe consequences of drunk driving on the one hand, and the need for user engagement on the other hand. They achieved this by two means. First, they decided to represent the game itself at a level of abstraction different from the effects of alcohol. In particular, the game implemented widely-known “Mario Kart”-style game mechanics and a simple, cartoonish look (Figure 2(a)), whereas the impairment of alcohol was reflected by realistic effects, including the deterioration of clear-sightedness, darkening the edges of the vision, attenuating sounds, and prolonged reaction times (realised by increased simulation speed), see Figure 3. Second, they introduced gamification elements including timed laps and collecting high scores by picking up precious diamonds from the track (Figure 2(b)). However, fundamental game mechanic to engage the users was invented later during the experimentation activity of the development phase. The students laid out the development phase very professionally and, together with the other students, received bi-weekly feedback to stay on track. Knowing that experiments could yield the key to an engaging user experience, the students tested various parameter settings of the driving model, its reactivity to the user input, as well as different interaction modes between the steered vehicle and the environment. From what they learned they were able to invent a mechanism to ensure a challenging and well-directed user experience. In particular, they translated the idea of collectibles on the track to their application domain and positioned beer cans at certain locations (Figure 2(b)). Their uptake would increase the blood alcohol level and driving would be impaired. The impairments would render it difficult to complete a track within a certain amount of time. Given the mechanics of driving, impaired driving, high-scores and time-laps, the students just needed to find the right balance to finish the development phase of their simulation. “Drink & Drive” was voted best entry in the public presentations at the end of last term’s interactive simulation course. In addition, it stirred a lot of excitement when it was offered for play as part of the Girls’ and Boys’ Day at our university. Based on these successes, the students feel that the most fundamental aspect that could drive a second development cycle would be the port of “Drink & Drive” to mobile devices for reaching a greater audience.

5 Conclusion and Future Work

In this paper, we presented an experience to adapt, teach and apply the CoSMoS process in a graduate computer science course on interactive simulation. We first laid out the multifaceted synopsis of the course before elaborating on the central role of the CoSMoS process in the context of the students’ term-long projects. Finally, we briefly presented some of the results of the students’ works and expanded on one of them, exemplarily. The scientific claim, the notion of self-organising processes with a focus on the interaction of numerous interwoven parts, as well as the agility of the CoSMoS process lend themselves well for backing interactive simulation projects.

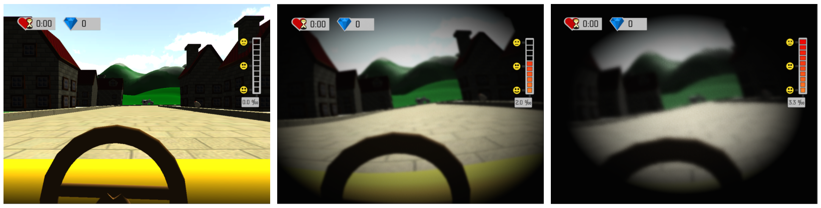


(a) (b)

Fig. 2. (a) A first-person default view is reduced to a simple steering wheel dashboard and a few icons that represent the time left to complete the track (the heart icon in the upper-left corner), the achieved score (the diamond icon next to the heart icon), and the alcohol blood concentration (to the right-hand side). (b) Alcoholic beverages and diamonds can be picked up from the road - the first increases the driver's blood alcohol concentration, the latter his score.

Although both the results and the students' feedback have been rather encouraging regarding the course contents, its layout and its general methodology, we are eager to further improve several aspects. It might, for instance, be beneficial to have certain activities of the different phases of the CoSMoS process take place in groups during the tutorial sessions. Scoping during the discovery phase has repeatedly proven difficult to students. An experienced teacher could guide the process and ensure that multiple options are considered by each group. More generally, we believe the CoSMoS process could still be more tightly integrated in both the lectures and the tutorials, by providing an outlook of its application to the lecture units' contents. For instance, one could illustrate the application of the CoSMoS phases not only to the project as a whole but also to individual aspects such as computer graphics and visualisation—from the goals and ideas of the used assets, the designed environment, over their creation and programming to experimenting with their parameters.

So far, we have not considered building on the CoSMoS process for evaluating the students' works or their performances during the exams, except for considering CoSMoS-supported project criteria (Section 3.1). Yet, the students frequently utilised the structure of the process for classifying and presenting their work. In particular, they frequently referred



0.0‰ 2.0‰ 3.3‰

Fig. 3. The alcohol blood level directly translates to impairments of vision, hearing, and reactivity.

to its phases and activities during their bi-weekly oral presentations and let their final project reports revolve around them. Hence, one research question that remains is whether and to which extent the individual phases of the CoSMoS process could be coupled a priori with the students' evaluation.

Last but not least, the CoSMoS process could be expanded to even better accommodate the development of interactive simulations. As they are typically designed for learning and training, an according 'engagement model' could, for instance, be an additional, desirable product of the discovery phase, complementing the domain model. It could comprise learning targets, explicitly visualised versus implicitly utilised data, the tasks and mechanics of the interfaces provided for interacting and exploring the domain model, as well as means of motivation, such as gamification elements. In combination with the domain model, such an engagement model would provide for a clear conceptual foundation for the development phase.

References

[1] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-time rendering.* CRC Press, 2011.

[2] P. S. Andrews, F. A. Polack, A. T. Sampson, S. Stepney, and J. Timmis, "The cosmos process version 0.1: A process for the modelling and simulation of complex systems," *Department of Computer Science, University of York, Tech. Rep. YCS-2010-453*, 2010.

[3] J. Banks, *Discrete Event System Simulation, 4/e.* Pearson Education India, 2005.

[4] J. Banks *et al.*, *Handbook of simulation.* Wiley Online Library, 1998.

[5] W. Banzhaf, "Artificial chemistries–towards constructive dynamical systems," *Solid State Phenomena*, vol. 97, pp. 43–50, 2004.

[6] R. W. Batterman, "Falling cats, parallel parking, and polarized light," *Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics*, vol. 34, no. 4, pp. 527–557, 2003.

[7] J. Bender, K. Erleben, J. Trinkle, and E. Coumans, "Interactive simulation of rigid body dynamics in computer graphics," *STAR Proceedings of Eurographics*, 2012.

[8] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems," in *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*. ACM, 2007, pp. 281–288.

[9] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović *et al.*, "Predicting protein structures with a multiplayer online game," *Nature*, vol. 466, no. 7307, pp. 756–760, 2010.

[10] J. Denzinger and C. Winder, "Combining coaching and learning to create cooperative character behavior." in *CIG*, 2005.

[11] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and D. Dixon, "Gamification. using game-design elements in non-gaming contexts," in *PART 2: Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*. ACM, 2011, pp. 2425–2428.

[12] A. Dix, *Human computer interaction*. Pearson Education, 2004.

[13] R. Diziol, J. Bender, and D. Bayer, "Robust real-time deformation of incompressible surface meshes," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2011, pp. 237–246.

[14] K. Erleben, J. Sporring, K. Henriksen, and H. Dohlmann, *Physics-based animation*. Charles River Media Hingham, 2005.

[15] R. Featherstone and D. Orin, "Robot dynamics: Equations and algorithms," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 826–834.

[16] J. D. Foley, V. L. Wallace, and P. Chan, "The human factors of computer graphics interaction techniques," *Computer Graphics and Applications, IEEE*, vol. 4, no. 11, pp. 13–48, 1984.

[17] B. Fry, *Visualizing data*. O'Reilly, 2008.

[18] A. Fuhrmann, G. Sobotka, and C. Groß, "Distance fields for rapid collision detection in physically based modeling," in *Proceedings of GraphiCon 2003*, 2003, pp. 58–65.

[19] C. Geiger, R. Fritze, A. Lehmann, and J. Stöcklein, "Hyui: a visual framework for prototyping hybrid user interfaces," in *Proceedings of the 2nd international conference on Tangible and embedded interaction*. ACM, 2008, pp. 63–70.

[20] J.-L. Giavitto, "Topological collections, transformations and their application to the modeling and the simulation of dynamical systems," in *Rewriting Techniques and Applications*. Springer, 2003, pp. 208–233.

[21] F. Groh, "Gamification: State of the art definition and utilization," *Institute of Media Informatics Ulm University*, pp. 39–47, 2012.

[22] I.-K. Hong, J.-B. Ryu, J.-H. Cho, K.-H. Lee, and W.-S. Lee, "Development of a driving simulator for virtual experience and training of drunk driving," in *3rd International Conference on Road Safety and Simulation*, 2011.

[23] H. Hoppe, "Smooth view-dependent level-of-detail control and its application to terrain rendering," in *Visualization'98. Proceedings.* IEEE, 1998, pp. 35–42.

[24] J. Hummel, R. Wolff, T. Stein, A. Gerndt, and T. Kuhlen, "An evaluation of open source physics engines for use in virtual reality assembly simulations," in *Advances in Visual Computing.* Springer, 2012, pp. 346–357.

[25] P. Humphreys, "Extending ourselves," *Science at Century's End: Philosophical Questions on the Progress and Limits of Science*, p. 13, 2004.

[26] P. J. Hunter and T. K. Borg, "Integration from proteins to organs: the physiome project," *Nature Reviews Molecular Cell Biology*, vol. 4, no. 3, pp. 237–243, 2003.

[27] A. Husselmann and K. Hawick, "Spatial data structures, sorting and gpu parallelism for situated-agent simulation and visualisation," in *Proc. Int. Conf. on Modelling, Simulation and Visualization Methods (MSV'12)*, 2012, pp. 14–20.

[28] J. A. Izaguirre, R. Chaturvedi, C. Huang, T. Cickovski, J. Coffland, G. Thomas, G. Forgacs, M. Alber, G. Hentschel, S. A. Newman *et al.*, "Compucell, a multi-model framework for simulation of morphogenesis," *Bioinformatics*, vol. 20, no. 7, pp. 1129–1137, 2004.

[29] A. Jaimes and N. Sebe, "Multimodal human–computer interaction: A survey," *Computer vision and image understanding*, vol. 108, no. 1, pp. 116–134, 2007.

[30] W.-K. Jeong, J. Beyer, M. Hadwiger, R. Blue, C. Law, A. Vázquez-Reina, R. C. Reid, J. Lichtman, and H. Pfister, "Ssecrett and neurotrace: interactive visualization and analysis tools for large-scale neuroscience data sets," *Computer Graphics and Applications, IEEE*, vol. 30, no. 3, pp. 58–70, 2010.

[31] M. M. Jones, "On-line simulation," in *Proceedings of the 1967 22Nd National Conference*, ser. ACM '67. New York, NY, USA: ACM, 1967, pp. 591–599. [Online]. Available: http://doi.acm.org/10.1145/800196.806028

[32] S. Kauffman, *The origins of order.* Oxford Univ. Press New York, 1993.

[33] R. Koster, *Theory of fun for game design.* O'Reilly Media, Inc., 2010.

[34] J. Krause, M. Spicker, L. Wörteler, M. Schäfer, L. Zhang, and H. Strobelt, "Interactive visualization for real-time public transport journey planning," in *SIGRAD*, 2012, pp. 95–98.

[35] T. Larsson and T. Akenine-Möller, "Collision detection for continuously deforming bodies." Eurographics, 2001.

[36] ——, "A dynamic bounding volume hierarchy for generalized collision detection," *Computers & Graphics*, vol. 30, no. 3, pp. 450–459, 2006.

[37] P. Lindemann, "The gilbert-johnson-keerthi distance algorithm," *Algorithms in Media Informatics*, 2009.

[38] M. Liu and G. Liu, "Smoothed particle hydrodynamics (sph): an overview and recent developments," *Archives of computational methods in engineering*, vol. 17, no. 1, pp. 25–76, 2010.

[39] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, "Unified particle physics for real-time applications," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 153:1–153:12, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601152

[40] M. S. McGuire and O. C. Jenkins, *Creating games: Mechanics, content, and technology*. AK Peters Limited, 2009.

[41] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, "Over two decades of integration-based, geometric flow visualization," in *Computer Graphics Forum*, vol. 29, no. 6. Wiley Online Library, 2010, pp. 1807–1829.

[42] B. A. Myers, "A brief history of human-computer interaction technology," *interactions*, vol. 5, no. 2, pp. 44–54, 1998.

[43] R. E. Nance, "A history of discrete event simulation programming languages," in *History of programming languages—II.* ACM, 1996, pp. 369–427.

[44] S. Narayanan and P. Kidambi, "Interactive simulations: History, features, and trends," in *Human-in-the-Loop Simulations.* Springer, 2011, pp. 1–13.

[45] R. O'Keefe, "Simulation and expert systems-a taxonomy and some examples," *Simulation*, vol. 46, no. 1, pp. 10–16, 1986.

[46] J. R. Parker, N. Sorenson, N. Esmaeili, R. Sicre, P. Gil, V. Kochlar, L. Shyba, and J. Heerema, "The booze cruise: Impaired driving in virtual spaces." *IEEE Computer Graphics and Applications*, vol. 29, no. 2, pp. 6–10, 2009.

[47] L. A. Pegden, T. I. Miles, and G. A. Diaz, "Graphical interpretation of output illustrated by a siman manufacturing system simulation," in *Proceedings of the 17th conference on Winter simulation.* ACM, 1985, pp. 244–251.

[48] C. A. Petri and W. Reisig, "Petri net," *Scholarpedia*, vol. 3, no. 4, p. 6477, 2008.

[49] F. A. Polack, P. S. Andrews, and A. T. Sampson, "The engineering of concurrent simulations of complex systems," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on.* IEEE, 2009, pp. 217–224.

[50] P. Prusinkiewicz, A. Lindenmayer, J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. de Boer, and L. Mercer, *The algorithmic beauty of plants.* Springer-Verlag New York, 1990, vol. 2, no. 6.

[51] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 25–34.

[52] L. Rothrock and S. Narayanan, *Human-in-the-loop Simulations: Methods and Practice.* Springer, 2011.

[53] A. Sarraf Shirazi, T. Davison, S. von Mammen, J. Denzinger, and C. Jacob, "Adaptive agent abstractions to speed up spatial agent-based simulations," *Simulation Modelling Practice and Theory*, vol. 40, pp. 144–160, 2014.

[54] A. Sarraf Shirazi, S. von Mammen, and C. Jacob, "Abstraction of agent interaction processes: Towards large-scale multi-agent models," *Simulation*, vol. 89, no. 4, pp. 524–538, 2013.

[55] A. R. Stage, N. L. Crookston, and R. A. Monserud, "An aggregation algorithm for increasing the efficiency of population models," *Ecological modelling*, vol. 68, no. 3, pp. 257–271, 1993.

[56] J. Steele and N. Iliinsky, *Beautiful visualization.* O'Reilly Media, Inc., 2010.

[57] T. Susi, M. Johannesson, and P. Backlund, "Serious games: An overview," 2007.

[58] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser *et al.*, "Collision detection for deformable objects," in *Computer Graphics Forum*, vol. 24, no. 1. Wiley Online Library, 2005, pp. 61–81.

[59] D. Van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *International Journal of Virtual Reality*, vol. 9, no. 2, p. 1, 2010.

[60] S. von Mammen and C. Jacob, "The evolution of swarm grammars-growing trees, crafting art, and bottom-up design," *Computational Intelligence Magazine, IEEE*, vol. 4, no. 3, pp. 10–19, 2009.

[61] S. von Mammen, J.-P. Steghöfer, J. Denzinger, and C. Jacob, "Self-organized middle-out abstraction," in *Self-Organizing Systems.* Springer, 2011, pp. 26–31.

[62] M. Ward, G. G. Grinstein, and D. Keim, *Interactive data visualization: Foundations, techniques, and applications.* AK Peters, 2010.

[63] A. H. Watt and F. Policarpo, *3D games: real-time rendering and software technology.* Addison-Wesley, 2001, vol. 1.

[64] S. Wendel and C. Dibble, "Dynamic agent compression," *Journal of Artificial Societies and Social Simulation*, vol. 10, no. 2, p. 9, 2007.

[65] P. C. Wong and R. D. Bergeron, "30 years of multidimensional multivariate visualization." in *Scientific Visualization*, 1994, pp. 3–33.

[66] M. Wooldridge, *An introduction to multiagent systems.* Wiley. com, 2008.

[67] M. M. Woolfson and G. J. Pert, *An introduction to computer simulation.* Oxford University Press Oxford, 1999.

[68] A. Wuensche, "Discrete dynamics lab," in *Artificial life models in software.* Springer, 2009, pp. 215–258.

Towards a Platform Model of the IL-1 Stimulated NF-κB Signalling Pathway using UML and Communicating Stream X-Machines

Richard A. Williams[1,2,3], Jon Timmis[3,4], and
Eva E. Qwarnstrom[5]

[1] Department of Management Science, Lancaster University, UK
`r.williams4@lancaster.ac.uk` (current address)
[2] Department of Computer Science, University of York, UK
[3] York Computational Immunology Laboratory, University of York, UK
[4] Department of Electronics, University of York, UK
[5] Department of Cardiovascular Science, Sheffield University Medical School, UK

The Nuclear Factor-kappa B (NF-κB) signalling pathway is one of the key signalling pathways involved in the control and regulation of the immune system [3]. Activation of the NF-κB transcription factor is a tightly regulated event, with NF-κB normally sequestered in the cytosol of non-stimulated cells. Following activation of a cell membrane receptor and propagation of the signal via intracellular signalling to the IκB Kinase (IKK), phosphorylation-induced degradation of IκB inhibitors occurs to facilitate the release of NF-κB and its translocation to the nucleus. Dysregulation of the pathway is known to be involved in a large number of inflammatory diseases.

Although considerable research has been performed since its discovery in 1986, we are still not in a position to control the signalling pathway, and thus limit the effects of NF-κB within promotion of inflammatory diseases. Through adherence to the CoSMoS framework, we are developing a computational model of the IL-1 stimulated NF-κB intracellular signalling pathway, to assist in promoting our understanding of the mechanistic behaviours within the signalling network, and therefore identify potential targets for therapeutic interventions. We have previously developed a separate *domain model* [4, 5] as advocated by the CoSMoS framework, which captures the essential processes and entities of the system under study using; in particular, the emergent behaviour, at an appropriate level of abstraction using a mixture of cartoon and UML diagrams, along with statistical techniques to define the temporal-spatial dynamics.

The next step in the CoSMoS framework is the development of a *platform model*, which details how the simulation is designed and provides an intermediate model that links the domain model to the forthcoming agent-based computational model (the *simulation platform*). We have developed our platform model, through the use of UML diagrammatic notations for modelling the high-level interactions between agents and the activities that they may perform; along with X-Machine mathematical notation, X-Machine diagrams and stategraph diagrams for modelling the low-level detailed specification of (programming language and architecture specific) interactions between agents, and the internal processing logic of individual agents.

As per our previous domain model, UML class diagrams were used in our platform model to represent the containment, inheritance and association characteristics of agents. The order of interactions within the system has again been documented through UML sequence, communication and activity diagrams. UML state machine diagrams were also used to express the detailed biological state changes of individual system components, however these were also complemented with X-Machine diagrams to express the detailed internal state changes of individual system components.

As found when developing the domain model, we believe that the activity diagram with swim-lanes has been the most useful notation for conveying the technical specifications of the system regarding the consequences of interactions between components, and that state machine diagrams are the most useful notation for defining the technical specification of individual system components. Unlike the domain model, the platform model also includes implementation specific details and as per [1] we have found it useful to document the various assumptions and constraints (regarding the technical scope of the computational model) as bullet points.

One of the key strengths of the CoSMoS process is the advocation of separating the abstracted view of biology (documented within the domain model) from the technical specification of the computational model (documented within the platform model). This separation ensures the abstracted view of biology and the technical specifications of the system remain discrete models, and thus aims to minimise confusion during the development of the computational model around what aspects of the programming code relate to biology requirements, and what aspects are necessary as technical workarounds due to constraints of the specific programming frameworks being used (e.g. communicating X-Machines and FLAME). As such, we believe the process of platform modelling to be an integral part of the development lifecycle for computational models of biological systems, and believe that our platform model will provide an unambiguous specification for the *simulation platform*, which will be developed using the FLAME simulation framework [2].

References

[1] P. S. Andrews, F. A. C. Polack, A. T. Sampson, J. Timmis, L. Scott, and M. C. Coles, “Simulating biology: Towards understanding what the simulation shows,” in *Proceedings of the 2008 Workshop on Complex Systems Modelling and Simulation*, S. Stepney, F. A. C. Polack, and P. Welch, Eds. Luniver Press, 2008, pp. 93–123.

[2] S. Coakley, “Formal software architecture for agent-based modelling in biology,” Ph.D. dissertation, University of Sheffield, 2007.

[3] S. Ghosh, M. J. May, and E. B. Kopp, “NF-κB and Rel proteins: Evolutionarily conserved mediators of immune response,” *Annual Review of Immunology*, vol. 16, pp. 225–260, 1998.

[4] R. A. Williams, “An agent-based model of the IL-1 stimulated nuclear factor-kappa b signalling pathway,” Ph.D. dissertation, University of York, 2014.

[5] R. A. Williams, J. Timmis, and E. E. Qwarnstrom, “Modelling complex dynamical biosystems using uml and statistical techniques,” In Preparation.