

Simulating Biology: Towards Understanding What the Simulation Shows

Paul S. Andrews¹ Fiona A. C. Polack¹ Adam T. Sampson²
Jon Timmis^{1,4} Lisa Scott³ Mark Coles³

¹Department of Computer Science, University of York, UK

²Computing Laboratory, University of Kent, UK

³York Centre for Immunology and Infection, University of York, UK

⁴Department of Electronics, University of York, UK

11th September 2008



- We are interested in studying complex systems
- We focus on two problems in this talk
 - 1 How to construct complex systems simulations
 - 2 Identifying how faithful simulation is to the biology: what does it tell you about the science



Complex systems

- A collection of component systems
 - Each component is a complete system
 - Components interact through environmental media
 - Components may be very simple or very complex
- Potential for emergent behaviours
 - From an outside perspective
 - Behaviours that are not simply the sum of the outputs of the component systems
- Homogeneous complex systems
 - Very large number of components
eg. cells
 - Small number of sorts of component



- A collection of component systems
 - Each component is a complete system
 - Components interact through environmental media
 - Components may be very simple or very complex
- Potential for emergent behaviours
 - From an outside perspective
 - Behaviours that are not simply the sum of the outputs of the component systems
- Homogeneous complex systems
 - Very large number of components
eg. cells
 - Small number of sorts of component



- A collection of component systems
 - Each component is a complete system
 - Components interact through environmental media
 - Components may be very simple or very complex
- Potential for emergent behaviours
 - From an outside perspective
 - Behaviours that are not simply the sum of the outputs of the component systems
- Homogeneous complex systems
 - Very large number of components
eg. cells
 - Small number of sorts of component



- *Model*: an abstraction to aid understanding or description
- A model presents many views of a system
 - Biologists and software engineers use diagrams to describe static structures and patterns of interaction (as Garnett et al this morning)
 - For a whole complex system, models need to describe features of component systems, high-level system, and environment
- Static modelling cannot describe many complex features:
 - Time, other than by annotation
 - Space, in that interaction only happens if components are physically contiguous
 - The features and consequences of many components operating simultaneously in an environment
 - Key features of the (dynamic) environment in which components exist



- *Model*: an abstraction to aid understanding or description
- A model presents many views of a system
 - Biologists and software engineers use diagrams to describe static structures and patterns of interaction (as Garnett et al this morning)
 - For a whole complex system, models need to describe features of component systems, high-level system, and environment
- Static modelling cannot describe many complex features:
 - Time, other than by annotation
 - Space, in that interaction only happens if components are physically contiguous
 - The features and consequences of many components operating simultaneously in an environment
 - Key features of the (dynamic) environment in which components exist



- Simulation provides time, space and environment
 - A complex system simulation can be thought of as executing the systems described in static models, many times, in parallel
 - Space can be included, and used in visual output
 - Components are implemented as a set of capabilities and responses cf. agents
 - Key features of the (dynamic) environment are simulated to provide the context for component behaviour and interaction cf. Harel et al's Reactive Animation



What does a simulation tell us?

- Complex systems are dynamical: small differences have potentially large results
- A significant concern in complex systems is the validity of simulation
- Simulations may use mathematical models
 - Differential equations, regression statistics
 - Can mimic observed trends, but poor choice of variables can give spurious equivalence
 - Little insight in to mechanisms and behaviours
- Agent-based simulation with appropriate numbers of simple components interacting freely in an environment
 - Selected details of components and environment has a significant effect on the simulation outcomes



What does a simulation tell us?

- Complex systems are dynamical: small differences have potentially large results
- A significant concern in complex systems is the validity of simulation
- Simulations may use mathematical models
 - Differential equations, regression statistics
 - Can mimic observed trends, but poor choice of variables can give spurious equivalence
 - Little insight in to mechanisms and behaviours
- Agent-based simulation with appropriate numbers of simple components interacting freely in an environment
 - Selected details of components and environment has a significant effect on the simulation outcomes



What does a simulation tell us?

- Complex systems are dynamical: small differences have potentially large results
- A significant concern in complex systems is the validity of simulation
- Simulations may use mathematical models
 - Differential equations, regression statistics
 - Can mimic observed trends, but poor choice of variables can give spurious equivalence
 - Little insight in to mechanisms and behaviours
- Agent-based simulation with appropriate numbers of simple components interacting freely in an environment
 - Selected details of components and environment has a significant effect on the simulation outcomes



Developing Complex Simulations

- Implementation involves conventional software engineering and verification of components
- Analysis and modelling is requirements capture from the real world
 - Requirements for components
 - Requirements for higher-level complex behaviours
 - Anti-requirements for things that the higher-level system must not do
- Experimentation explores how the simulation relates to reality
- Complex system simulation must also represent relevant parts of the environment



Developing Complex Simulations

- Implementation involves conventional software engineering and verification of components
- Analysis and modelling is requirements capture from the real world
 - Requirements for components
 - Requirements for higher-level complex behaviours
 - Anti-requirements for things that the higher-level system must not do
- Experimentation explores how the simulation relates to reality
- Complex system simulation must also represent relevant parts of the environment



An Example Complex System Simulation

- Over to Paul → → → → → → → → → → → → → → →



- The immune system is a complex system
- Develop a model and simulation of an immune mechanism
- Background biology first



An Example Complex System Simulation

- Over to Paul → → → → → → → → → → → → → → →

-



- The immune system is a complex system
- Develop a model and simulation of an immune mechanism
- Background biology first



An Example Complex System Simulation

- Over to Paul → → → → → → → → → → → → → → →

-



- The immune system is a complex system
- Develop a model and simulation of an immune mechanism
- Background biology first



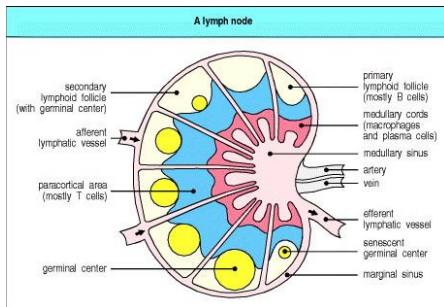
The Immune System

- Immune system provides defence against pathogens (viruses, bacteria, parasites)
- Comprises a variety of cells, molecules and organs
- Leukocytes (white blood cells) are the cells specific to the immune system
- Lymphocytes are a subset of leukocytes vital to the immune response
 - These are the cells we are interested in

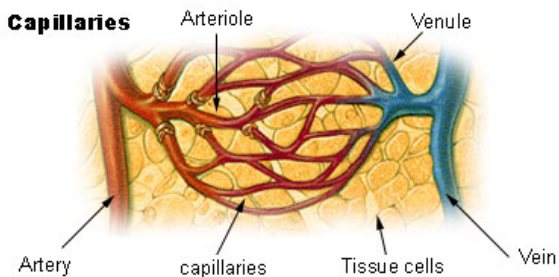


Lymph Nodes

- Immune organs where immune responses initiated and antibodies produced
- Hundreds throughout the body
- Cells enter through blood or lymphatic system

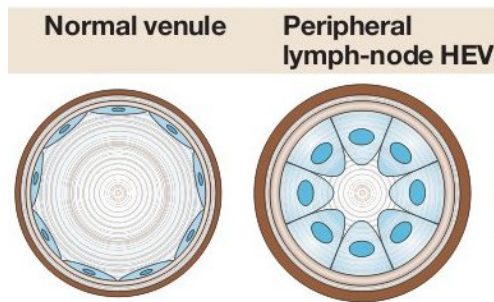


- Venules bring de-oxygenated blood to the veins from capillary bed

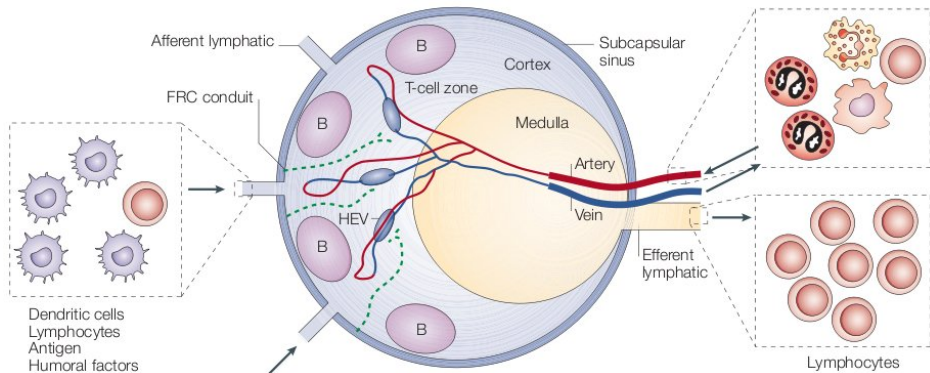


High Endothelial Venules (HEVs)

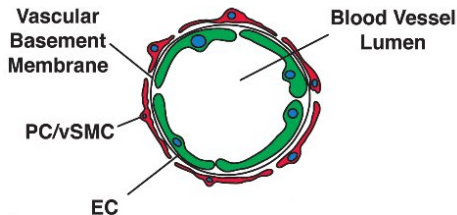
- Certain areas of the lymph node venule network are made up of HEVs
- HEVs characterised by tall and plump endothelial cells



High Endothelial Venules In Lymph Node

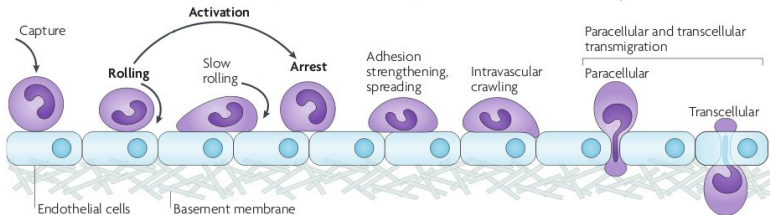


- Cells that wrap around small blood vessels
 - Act as scaffolding
 - Similar to smooth muscle cells
- Constriction and dilation regulates diameter and blood flow of vessel



Lymphocyte Rolling

- Lymphocytes enter lymph node through HEVs
 - Initiate in a rolling process
 - Under certain conditions, lymphocytes slow and squeeze though between endothelial cells



A Purpose for the simulation

- There should be a reason for what we are doing
- Implement something that is *biologically faithful*
- Aid hypotheses testing
 - E.g. *the increase in lymphocytes in the lymph node during infection is due to dilation of the high endothelial venules*
- Desired output
 - Numerical data under different conditions
 - A format that allows comparison with *in vitro* experimental data



A Conceptual Model

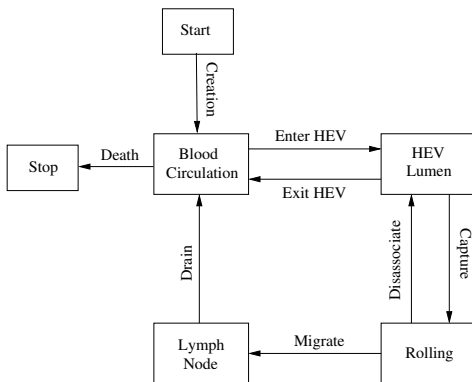
- Population of homogeneous lymphocytes interacting in an environment
- Environment:
 - Parts of the body with which the lymphocytes interact
 - Tube (HEV) consisting of HE cells and pericytes form of a tube
- We model different environments lymphocytes pass through as *states*
- *Transitions* occur when a lymphocyte moves between environments



State diagram model for a lymphocyte

Start: lymphocyte is 'born'

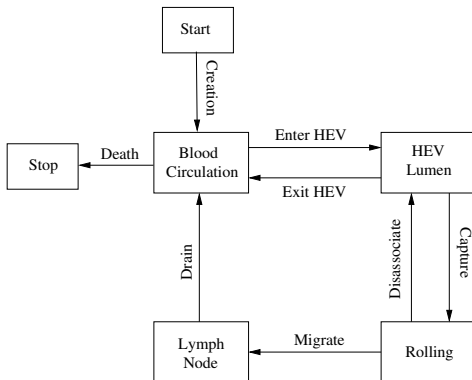
- *Creation* transits to **Blood circulation**



State diagram model for a lymphocyte

Blood Circulation: parts of the body that the lymphocyte is in when it is not in the HEV or the lymph node tissue

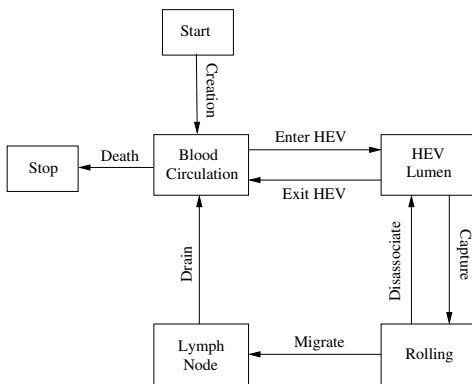
- *Enter HEV* transits to **HEV Lumen**
- *Death* transits to **Stop**



State diagram model for a lymphocyte

HEV Lumen: lymphocyte when it is flowing freely in the lumen of a HEV

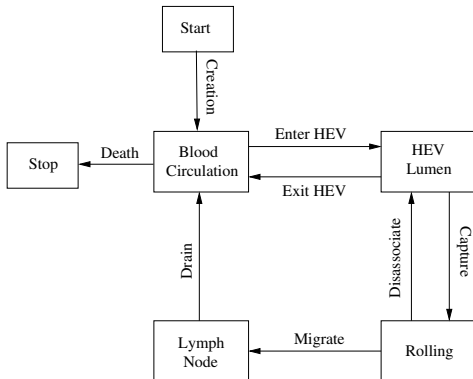
- *Exit HEV* transits to **Blood circulation**
- *Capture* transits to **Rolling**



State diagram model for a lymphocyte

Rolling: This state represents the lymphocyte when it is rolling on the interior surface of an HEV

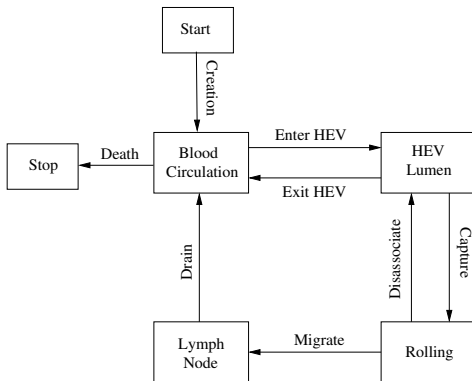
- *Disassociate* transits to **HEV Lumen**
- *Migrate* transits to **Lymph Node**



State diagram model for a lymphocyte

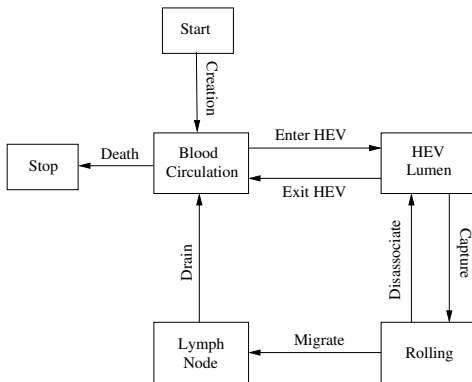
Lymph Node: This state describes the lymphocyte when it is present in functional tissue of a lymph node

- *Drain* transits to **Blood circulation**



State diagram model for a lymphocyte

Stop: lymphocyte 'dies'

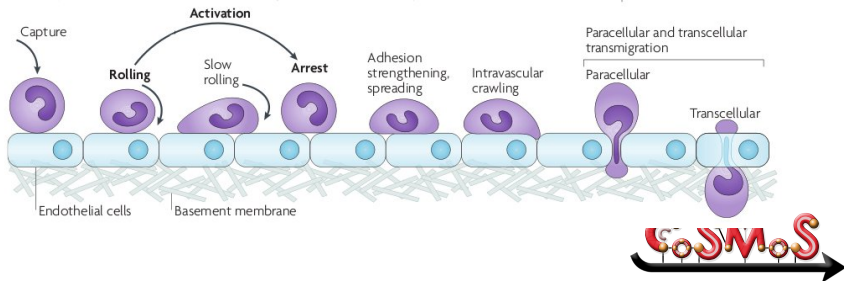


- Can annotate states and transition of model with assumptions
- Transition example: *Enter HEV*
 - The HEVs are homogeneous: The endothelial cells and pericytes that make up the HEV all behave the same making the HEV appear the same at all points.
 - Lymphocytes flow though the HEV at the same rate.
- More high-level assumptions also



Main simplification

- Reduce the multi-stage adhesion cascade down to two main steps
 - ① Capture of lymphocytes on to the endothelial wall
 - ② Migration after receiving the chemokine signal
- Other stages in the cascade are assumed to be either deterministic, or have such small probabilities of failing that they are insignificant
 - Note the assumption here!



- Developed two simulations of the conceptual model
- *Migration-abstract*
 - No explicit co-ordinate system, only the four body locations
 - Each of these four state spaces can contain a number of lymphocyte agents
- *Migration-space*
 - 3-dimensional HEV tube made up of endothelial cells,
 - Supports visualisation of the HEV and of the lymphocytes migration
 - Simulation is “closer to the biology”

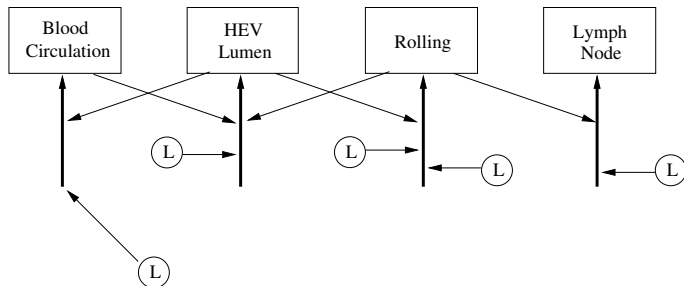


- Focus on this simulation
- Implement the lymphocyte state transition model for a population of lymphocytes
- Lymphocytes transit between locations
 - Follow only allowed transitions in model
 - Each transition has a probabilities
- Transition probabilities extracted from the biological detail.
- Documenting and annotating where this detail comes from *important*
- Implemented using *occam- π*
- Process-oriented programming (POP) language capable of massive concurrency



Lymphocyte and state processes

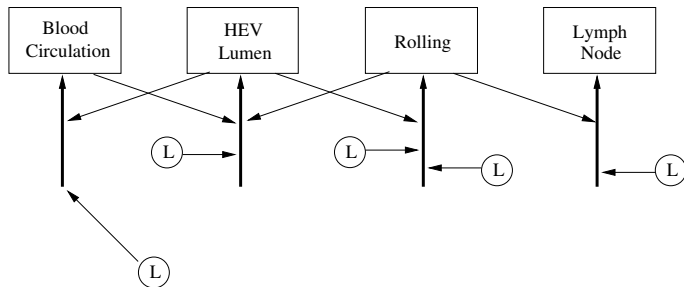
- Each lymphocyte agent is represented as a *process*
- Lymphocytes connected to one of the four body place states by a *communication channel*



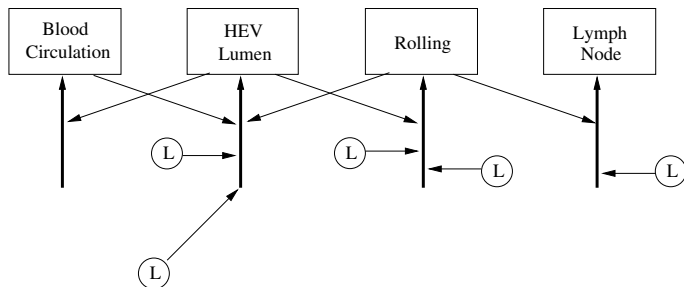
- Many thousands of lymphocyte processes in order to get close to the biological scale
- Each lymphocyte updates once per iteration (achieved via *occam-pi* barrier construct)
 - Lymphocyte tests for its possible for transitions
- Example: Lymphocyte in **rolling** state
 - Possible transitions: *disassociate* or *migrate*
 - Generate random number
 - If number falls in the range of a transition, then transit.
 - Else stays in its current state.



Lymphocyte movement between states



Lymphocyte movement between states



- A mapping exists between simulation probabilities and the biological detail
- For *confidence* in the simulation, need to validate that they represent what they are supposed to
- Exact biological details can be poorly understood or simply not recorded.
- Details can come from many different sources, based on different experiments using different technologies and subjects and species!
- In an ideal world, it is the job of a domain expert/s to validate the biological detail used is correct
- Ongoing work to find good probabilities



- Each simulation will require *seperate* validation
- Explicitly representing space can *aid* simulation
 - Removes having to consider location in the probability constuction: *capture* transition
- We must expose our validation argument to external review (domain expert)
- The connotations of validating the *capture* transition is investigated in more detail in the paper



Validation and assumptions

- Validation can reveal data inconsistencies and mismatches of assumptions
 - E.g. discredited research data has been used
- Clearly presenting assumptions and data sources leaves model/simulation open to validation
 - Collect assumptions and biological detail *together* e.g. tables
 - *Annotate* models and simulations with this detail
 - Provides traceability
 - Allows inspection by *domain expert*
- Most incorrect assumptions won't invalidate work, but require updating



- Back to Fiona → → → → → → → → → → → → → → → →

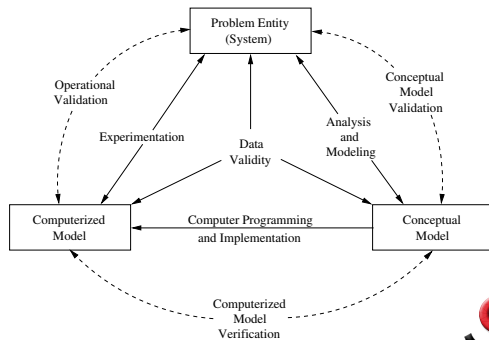


- Back to Fiona → → → → → → → → → → → → → → → →
- ↓
↓
↓
↓



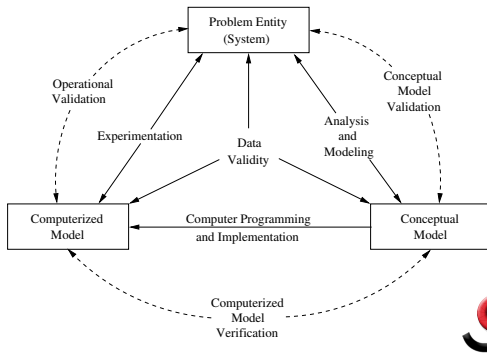
Non-Complex Simulation Validation

- Validation determines that the right system has been built
- Sargent's model validates:
 - conceptual model against reality
 - executing simulation against reality
 - data used in execution

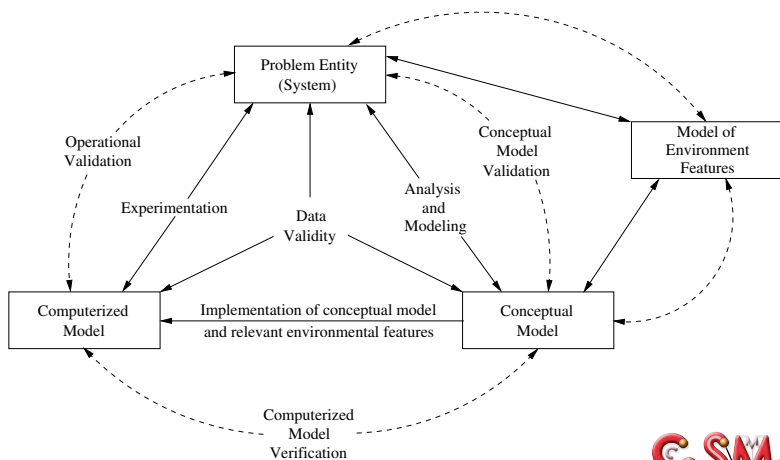


Non-Complex Simulation Validation

- Sargent and others propose many ways to validate
 - Compare the simulation to other valid models
 - Compare simulation outputs to historical or projected data
 - Validate simulation behaviour against processes in real systems
- The simulation can be shown to be a valid model of reality



Extending Sargent's Development Process to Complex System



- Complex systems behaviour is critically dependent on context: environment, number and state of component systems, etc
 - If the environment is slightly modified, or the quantity and distribution of components is changed, different behaviour may result
- Validation of a complex system cannot be absolute
 - The only way to faithfully construct the environment is a simulation of the Universe...

... which is a bit extreme



- Complex systems behaviour is critically dependent on context: environment, number and state of component systems, etc
 - If the environment is slightly modified, or the quantity and distribution of components is changed, different behaviour may result
- Validation of a complex system cannot be absolute
 - The only way to faithfully construct the environment is a simulation of the Universe...

... which is a bit extreme



Arguments of Validity

- High integrity systems engineering already relies on arguments of validity
 - Assurance of aircraft, nuclear controllers etc
 - Methods to create, structure, and check assurance arguments
 - certification depends on demonstration that risk is as low as reasonably possible
- Alexander 2007 extends assurance to development of complex systems using simulation
- We present initial ideas for recording simulation modelling assumptions etc to develop a validity argument
- A similar approach is used to document validity of data used to run Reactive Animation simulations (Harel et al.)
- The principle is to expose all assumptions and justifications to external scrutiny



Arguments of Validity

- High integrity systems engineering already relies on arguments of validity
 - Assurance of aircraft, nuclear controllers etc
 - Methods to create, structure, and check assurance arguments
 - certification depends on demonstration that risk is as low as reasonably possible
- Alexander 2007 extends assurance to development of complex systems using simulation
- We present initial ideas for recording simulation modelling assumptions etc to develop a validity argument
- A similar approach is used to document validity of data used to run Reactive Animation simulations (Harel et al.)
- The principle is to expose all assumptions and justifications to external scrutiny



Concluding Remarks: Modelling

- Model-driven development and other rigorous approaches are suitable for engineering components of complex systems
 - Some domain-specific adaptation is likely to be needed
 - Process-oriented approaches may be more appropriate than traditional OO
 - But verification is essentially the same as in conventional systems engineering
- Current modelling approaches do not capture time, space and environment:
 - Time and space can be expressed using simulation
 - Approaches that drive static models with realistic data are promising (eg. Reactive Animation)
 - "Data" defines initialisation of and evolutionary paths through static models



Concluding Remarks: Modelling

- Model-driven development and other rigorous approaches are suitable for engineering components of complex systems
 - Some domain-specific adaptation is likely to be needed
 - Process-oriented approaches may be more appropriate than traditional OO
 - But verification is essentially the same as in conventional systems engineering
- Current modelling approaches do not capture time, space and environment:
 - Time and space can be expressed using simulation
 - Approaches that drive static models with realistic data are promising (eg. Reactive Animation)
 - "Data" defines initialisation of and evolutionary paths through static models



Concluding Remarks: Validation

- Traditionally, validation checks the implementation against the requirements
- Complex systems can only be partially validated by such means
- Validity would require demonstration of emergent behaviours:
 - No definitive proof can be made that the required behaviours always emerge in the relevant conditions
 - No definitive proof can be made that any deprecated behaviours do not occur
- Instead, validation needs to present the assumptions behind the model [simulation] and expose these to critical scrutiny



Concluding Remarks: Validation

- Traditionally, validation checks the implementation against the requirements
- Complex systems can only be partially validated by such means
- Validity would require demonstration of emergent behaviours:
 - No definitive proof can be made that the required behaviours always emerge in the relevant conditions
 - No definitive proof can be made that any deprecated behaviours do not occur
- Instead, validation needs to present the assumptions behind the model [simulation] and expose these to critical scrutiny



- Fiona A. C. Polack, Tim Hoverd, Adam T. Sampson, Susan Stepney, and Jon Timmis. Complex systems models: Engineering simulations. In *Proceedings, ALife*, MIT Press, 2008 (online).
- F. Polack, S. Stepney, H. Turner, P. Welch, and F. Barnes. An architecture for modelling emergence in CA-like systems. In *Proceedings, ECAL*, LNAI Vol. 3630, Springer, 2005.
- R Alexander, M Hall-May, T Kelly, Characterisation of Systems of Systems Failures. In *Proceedings, Int. System Safety Conference*, System Safety Society, 2004.
- R. Alexander. *Using Simulation for Systems of Systems Hazard Analysis*. PhD thesis, Department of Computer Science, University of York, 2007.
- R. G. Sargent. An exposition on verification and validation of simulation models. In *Proceedings, Winter Simulation Conference*, ACM, 1985. (And subsequent Proceedings.)
- D. Harel, S. Efroni and I.R. Cohen, Reactive Animation, In *Proceedings, FMCO*, LNCS Vol. 2852, Springer, 2003. (And many other papers.)

