Joint Defra/EA Flood and Coastal Erosion Risk Management R&D Programme

Development and Demonstration of Systems Based Estuary Simulators (EstSim)

EstSim Methods and Software Tools for Estuary Behavioural System Simulation

R&D Project Record FD2117/PR4

Environment Agency

defra
Department for Environment
Food and Rural Affairs

Defra/Environment Agency
Flood and Coastal Defence R&D Programme

Development and Demonstration of Systems Based Estuary Simulators (EstSim)

Conceptualisation Report

Prepared by University College London for the Estuaries Research Programme (ERP Phase 2) within the Defra and Environment Agency Joint Broad Scale Modelling Theme

Record No: FD2117/PR4

May 2007

Publishing organisation

Environment Agency
Rio House
Waterside Drive
Aztec West
Almondsbury
Bristol BS32 4UD
Tel: 01454 624400 Fax: 01454 624409

Statement of Use
This document provides information for the Department for Environment Food and Rural Affairs (Defra) and Environment Agency Staff (EA), researchers and consultants about the research project Development and Demonstration of Systems Based Estuary Simulators (EstSim) and constitutes an R&D output from the Joint Defra/Environment Agency Flood and Coastal Defence R&D Programme (Estuaries Research Programme Phase 2).

Contract Statement
This report describes work commissioned by Defra under Project FD2117 Development and Demonstration of Systems Based Estuary Simulators (EstSim). The Funders Nominated Project Officer was Kate Scott (Environment Agency: kate.scott@environment-agency.gov.uk). The ABP Marine Environmental Research Ltd (ABPmer) Project Number was R/3434 and the Project Manager at ABPmer was Alun Williams (Email: awilliams@abpmer.co.uk).

Collaboration Statement
This report was prepared by the EstSim Consortium comprising: ABP Marine Environmental Research Ltd (lead), University of Plymouth (School of Engineering), University College London (Coastal & Estuarine Research Unit), HR Wallingford, WL | Delft Hydraulics and Discovery Software.

Disclaimer
This report is a contribution to research generally and it would be imprudent for third parties to rely on it in specific applications without first checking its suitability.

Various sections of this report rely on data supplied by or drawn from third party sources. The EstSim Consortium accept no liability for loss or damage suffered by the client or third parties as a result of errors or inaccuracies in such third party data.

The EstSim Consortium will only accept responsibility for the use of its material in specific projects where it has been engaged to advise upon a specific commission and given the opportunity to express a view on the reliability of the material for the particular applications.

Publication implies no endorsement by the Department for Environment, Food and Rural Affairs of the report's conclusions or recommendations.

Keywords
Behaviour, Estuary, Geomorphology, Modelling, Morphology.

Further copies of this report are available from:
Environment Agency R&D Dissemination Centre, c/o
WRc, Frankland Road, Swindon, Wilts SN5 8YF

tel: 01793-865000 fax: 01793-514562 e-mail: publications@wrcplc.co.uk

EXECUTIVE OVERVIEW OF FD2117: DEVELOPMENT AND DEMONSTRATION OF SYSTEMS BASED ESTUARY SIMULATORS (EstSim)

Methods and software tools for estuary behavioural system simulation, May 2007

Purpose

The Broad Scale Modelling Theme of the Defra/EA Joint Thematic R&D Programme for Flood & Coastal Defence has funded three contracts under the Estuaries Research Programme, Phase 2 (FD2107, FD2116 and FD2117). FD2117 (EstSim) started in April 2004 and has the following headline aims:

- To extend the ability to simulate estuary response to change.
- Facilitate knowledge exchange through accessibility of simulation results.

The Project

ABPmer, University College London, University of Plymouth, WL | Delft Hydraulics and Discovery Software are undertaking the project. The project was originally of 3 years duration (April 2004 – April 2007), but had an extension for completion in June 2007 and has nine Scientific Objectives as follows:

1. System Conceptualisation: Boundary setting and focusing of research effort.
2. Development of Management Questions: Rationalisation of management questions that can be informed through application of systems approach.
3. Development of Behavioural Statements: Formal definition of estuarine system in terms of systems approach and behavioural statements.
4. Mathematical Formalisation: Development of behavioural statements into a logically consistent mathematical framework.
5. Development of System Simulation: Development of architecture for estuary simulation based on the mathematical formulation of the system definition.
6. Manager System Interface: Explore the use of decision support systems and visualisation techniques for proof of concept testing.
7. Pilot Testing: Performance evaluation of estuary simulator.
8. Dissemination: Increase awareness of function and utility of research.
9. Peer Review: Ensure research lines deliver against Scientific Objectives.

This report follows on from the Mathematical Formalisation (Objective 4).
and further develops the mathematical framework into a system - the System Simulation (Objective 5).

In developing system simulation a number of stages have been undertaken, including:

- Qualitative modelling of estuary system behaviour.
- Computational aspects of behavioural system models.
- Software platforms for estuary behavioural system modelling.
- Alternative architectures for an EstSim software tool.
- Boolean network estuary behavioural system model.
- Conclusions and recommendations.

This report will be used to guide the development of the case studies and the simulator itself.

Contact Details

For more details please contact the FD2117 Project Manager Alun Williams (awilliams@abmer.co.uk) or the Funders' Nominated Project Officer Kate Scott (kate.scott@environment-agency.gov.uk).

CONTENTS

Tables

Figures

Appendices

1. INTRODUCTION

1.1 Background

On 1st April 2004 ABP Marine Environmental Research Ltd (ABPmer) and its Project Partners were awarded research contract FD2117 (CSA 6064) within the Broad Scale Modelling Theme of the Defra/EA Joint Thematic R&D Programme for Flood & Coastal Defence.

The contract for FD2117 was awarded on the basis of a 'contract won in competition' after submission of a CSG7 (revised CSG7 submitted on 8th March 2004).

Entitled 'Development and Demonstration of Systems-Based Estuary Simulators' (hereafter EstSim), this research contract forms one of three contracts awarded under Phase 2 of the Estuary Research Programme (ERP). The two other contracts under the umbrella of ERP Phase 2 are (i) FD2107: Development of Estuary Morphological Models, and (ii) FD2116: Review and Formalisation of Geomorphological Concepts and Approaches.

The three phases of the Estuaries Research Programme seek to improve our understanding and prediction of estuarine morphological change over the medium to long-term, thereby facilitating strategic and sustainable decisions regarding flood and coastal defence.

The EMPHASYS Consortium undertook Phase 1 of this programme by evaluating existing morphological modelling approaches with the most promising of these approaches being developed within ERP Phase 2. It is anticipated that Phase 3 will seek to incorporate prior ERP research into an 'Integrated Estuary Management System'.

1.2 Project Aims

The overall aim of EstSim is to extend the ability to simulate estuarine response to change. This will be achieved through the delivery of research into the systems-based approach as an alternative yet complementary methodology to those research lines being undertaken within the other ERP Phase 2 projects (morphological concepts, bottom-up, top-down and hybrid methods). EstSim will also explore the simulation process in order to facilitate knowledge exchange between the systems-based tools and estuary managers. Integration of the systems based approach and existing methods is shown conceptually within Figure 1.

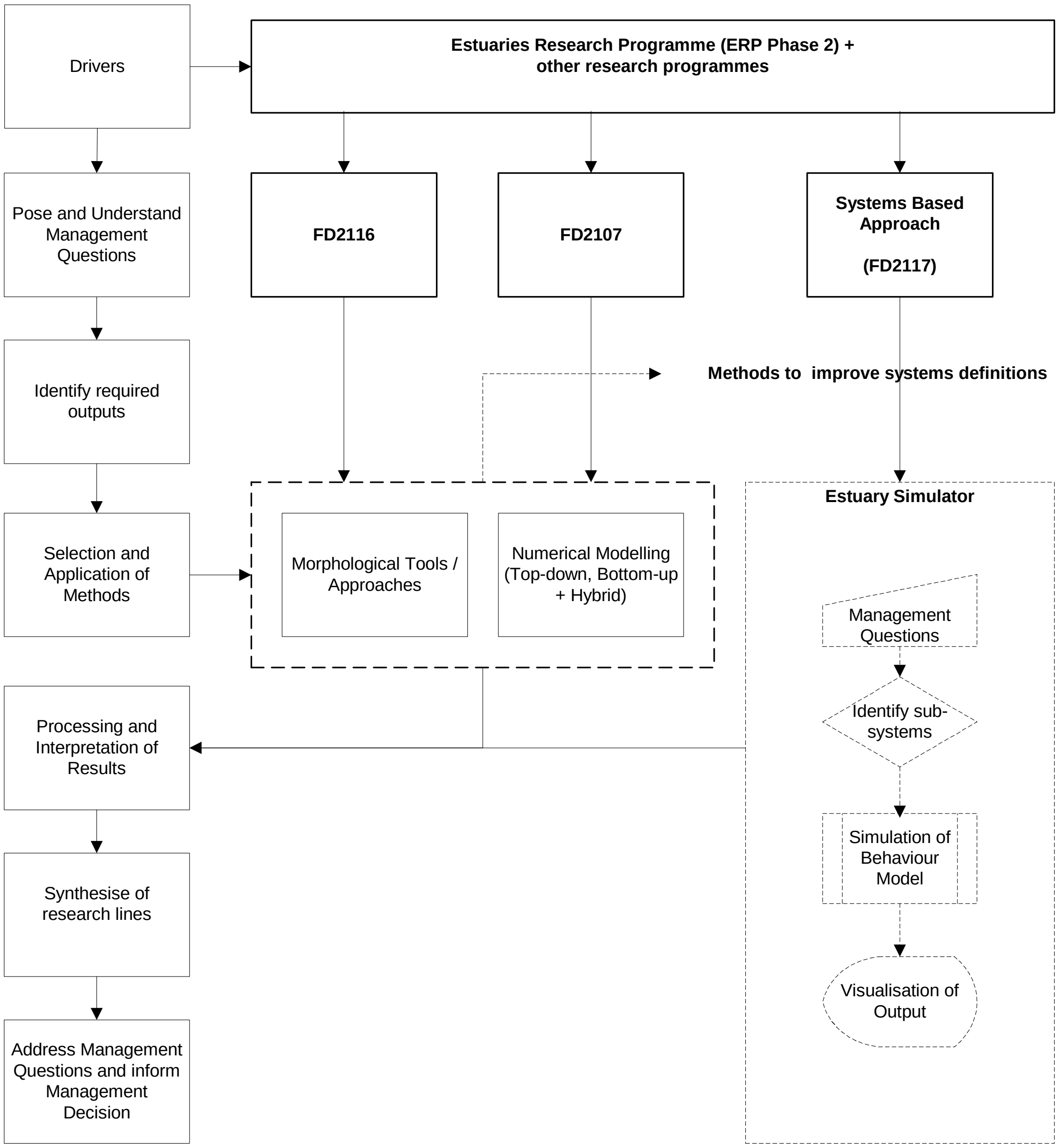


Figure 1. Integration of systems based approach

1.3 Project Structure

The project has been structured in to nine Scientific Objectives, covering the required lines of research and dissemination:

1. System Conceptualisation: Boundary setting and focusing of research effort.
2. Development of Management Questions: Rationalisation of management questions that can be informed through application of systems approach.
3. Development of Behavioural Statements: Formal definition of estuarine system in terms of systems approach and behavioural statements.
4. Mathematical Formalisation: Development of behavioural statements into a logically consistent mathematical framework.
5. Development of System Simulation: Development of architecture for estuary simulation based on the mathematical formulation of the system definition.
6. Manager System Interface: Explore the use of decision support systems and visualisation techniques for proof of concept testing.
7. Pilot Testing: Performance evaluation of estuary simulator.
8. Dissemination: Increase awareness of function and utility of research.
9. Peer Review: Ensure research lines deliver against Scientific Objectives.

1.4 Project Progress

Scientific Objective 3 was delivered through production of the EstSim Mathematical Formulation Report (PR3). This defined the mathematical structure taken forward to assist in the system design and simulation (PR4).

Following the completion of PR3, work was commenced on Objective 5.

1.5 System Simulation (Objective 5)

The objective of this research element is to set-up the architecture and methodology for estuary simulation based on the system definition comprising the behavioural statements its mathematical formulation. The simulation phase thereby provides the system approach output to inform answers to the specific management questions.

This Objective will include sensitivity testing and validation of the estuary simulator.

In order to deliver this research a number of sub-tasks were defined, these are given in Table 1.

Delivery of this final Technical Report effectively completes all tasks required under Objective 5, as listed in Table 1. This document constitutes Project Record PR4 and as such is submitted in as Primary Milestone 05/02.

Table 1. Objective 5 tasks

Task	Description
5.1	Critical review of simulation methods and tools (e.g. Stella, Modelmaker, Powersim, Matlab, Simulink and Extend) and choice of most appropriate simulation method to be applied.
5.2	Translation of Systems output (Objective 3 and 4) into estuary simulator.
5.3	Sensitivity testing and validation of simulator against development estuary properties.
5.4	Hold team workshop to disseminate findings.
5.5	Produce Technical reports on findings (code listings, user manuals etc.), and perform assessment of how well the resultant simulator addresses the management questions posed at the outset (Milestone 05/02).
5.6	Demonstrate simulator in parallel with Objective 6 at Technical Stakeholder Group meeting.

1.6 Follow-on Research

The work carried out within the System Simulation and Validation Objective has enabled the manager- system interface to be extended (Objective 6) and the Pilot Testing to commence, Objective 7.

1.7 Report Structure

The report has been structured to capture the main tasks but reordered for ease of presentation and review, as follows:

Section 2:	Qualitative modelling of estuary system behaviour
Section 3:	Computational aspects of behavioural system models
Section 4:	Software platforms for estuary behavioural system modelling
Section 5:	Alternative architectures for an EstSim software tool
Section 6:	Boolean network estuary behavioural system model
Section 7:	Conclusions and recommendations
Section 8:	The Way Forward (Future Objectives)
Appendix 1:	GUI based dynamics system simulation software product overview
Appendix 2:	Numerical computation and visualisation software product overview
Appendix 3:	Main morphological components for EstSim generic estuary types
Appendix 4:	Example of steering file format for prototype simulator

2. QUALITATIVE MODELLING OF ESTUARY SYSTEM BEHAVIOUR

As is the case generally in geomorphological systems, estuarine and coastal morphodynamic behaviour is complex because of the feedbacks between morphology and sediment transport. An important consequence of these feedbacks is that the operation of a system at any time is influenced by previous states or antecedent conditions (geomorphologists refer to this as *state dependence*, or *inheritance*). Additional complexity arises from the interplay between self-regulation (or *equilibrium tendency*) and self-forcing (which leads to *thresholds* and *complex response*), and the non-linear nature of many of the functional linkages between system components (see, for example, Wright and Thom, 1977; Cowell and Thom, 1994). Equilibrium represents a morphodynamic state that is stable for a given set of environmental boundary conditions. This stability can take the form of a steady state or oscillation about a long-term average condition. Many geomorphological systems can also potentially exist in a chaotic equilibrium (Phillips, 1992) characterised by complex aperiodic deterministic behaviour, albeit with some degree of pattern discernable through averaging of successive states.

The identification of equilibrium conditions and the manner in which estuary morphology adjusts towards a new equilibrium state following a change or perturbation in system operation are of great interest to estuary managers and users. In engineering, much use has been made of so-called *regime models*, which determine directly the stable morphology that arises from a balance between sediment transporting forces. These states can be determined analytically or numerically (in the case of some shoreline models) or empirically (e.g. tidal inlet scaling relationships of the kind developed by O’Brien (1969) and re-evaluated by Townend (2005)). These are effectively top-down models in the sense used in the ERP (EMPHASYS, 2000). Alternatively, the time-evolution of coastal or estuary morphology can be modelled explicitly. This is typically attempted through *process-based morphodynamic models* (bottom-up models in the sense of the ERP). These contain detailed mathematical formulations for hydrodynamics and sediment dynamics and are computationally intensive. Accordingly, they are usually executed in some form of time-stepping framework (e.g. de Vriend *et al.*, 1993) in which the time-step for morphological change is much greater than that for the simulation of physical processes (this approach is being taken forward in a related ERP2 Project, FD2107 ‘Development of Estuary Morphological Models’). Process-based morphodynamic models provide considerable insight into short-term system behaviour, but evolution towards a longer term and/or larger scale equilibrium is often incorrectly reproduced (e.g. Hibma *et al.*, 2004).

None of the above approaches are suited to the modelling of complex systems for which the functional relationships between morphological components and driving processes cannot all be mathematically specified in a way that is physically realistic. In the case of estuaries, quantitative functional representations are not available for many of the linkages and feedbacks between system components, and changing environmental forcing (e.g. sea-level and sediment supply) and constraints (e.g. resistant geology or engineered structures) may complicate the evolution of morphology and the associated process regime towards equilibrium.

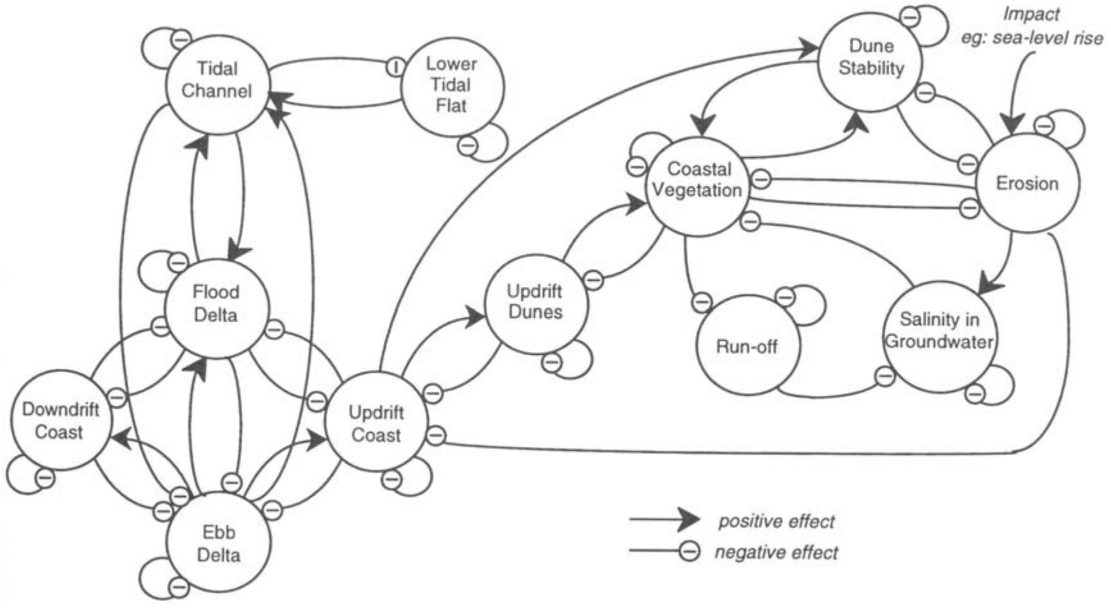
A qualitative approach addresses many of these problems and provides an alternative basis for modelling the behaviour of estuary and coastal systems, abstracted at the scales of most relevance to the tackling of estuary management questions (i.e. the ‘engineering scale’ of Cowell and Thom (1994) and the ‘meso-scale’ of Townend (2003)). A formal systems-based approach to the conceptualisation of estuarine and coastal systems, supported by the technical vocabulary of general systems theory (von Bertalanffy, 1968; Chorley, 1962; Bennett and Chorley, 1978), has been advocated by Townend (2003). The potential of qualitative modelling as a means of providing indicative rather than strictly quantitative insights into the behaviour of systems specified in this manner has been highlighted by Capobianco *et al.* (1999).

Capobianco *et al.* (1999) identify seven stages associated with the development of a qualitative model of system behaviour (Figure 2). These include the identification of qualitative variables and the causal linkages between them, which typically involves the construction of system diagrams of some form. Modelling of system behaviour then requires the definition of a mathematical quantity space to represent interaction between the state variables. Figure 3 shows a generic tidal inlet system subject to a rise in sea level, wherein the quantity space is defined in terms of a signed graph, with positive or negative effects connecting the main system variables. Transfer rules and knowledge formalisation are then needed to convert either quantitative understanding (which might take the form of empirical scaling relationships or physically-based process laws) or linguistic understanding (e.g. a statement describing the behaviour of a sub-component) of system linkages into a set of cause-effect relationships. Qualitative calculus here refers to the employment of mathematical analyses that treat the system as an interconnected network (or graph) in order to achieve insights into the direction of change in the state of any or all of the system variables in response to external forcing or an imposed alteration to system structure.

1. *Identification of qualitative variables* and the likely relationships between them
2. *Conceptual representation* and definition of causal inference network
3. *Definition of a quantity space* – a finite set of symbols to define values of state variables
4. *Establish transfer rules* that can be used to develop qualitative relationships between variables
5. *Knowledge formalisation*, in which transfer rules are used to define cause-effect relationships
6. *Qualitative translation of measurements*, where variables are assigned values from the quantity space
7. *Application of qualitative calculus* to analyse characteristics of the model

(Adapted from Capobianco *et al.*, 1999)

Figure 2. Construction of a qualitative model



(Modified from Capobianco *et al.*, 1999)

This is probably the simplest form of model, in which system inter-linkages are represented merely as directed influences (i.e. in the form of a 'signed graph').

Figure 3. Example of a qualitative model of the impact of sea-level rise on a hypothetical tidal inlet

As noted in the ERP Phase 2 Research Plan (French *et al.*, 2001), top-down qualitative modelling of this kind may provide useful knowledge of estuary behaviour at the scales most relevant to managers and users. New emergent properties of system behaviour may be revealed, including the existence of unexpected sensitivities to change and/or constraints upon the evolution towards an equilibrium morphology. Qualitative judgements concerning estuary behaviour also provide a basis for evaluating the output from quantitative models (i.e. in terms of the direction of predicted change rather than its magnitude). Such insights can be crucial in impact and vulnerability assessment.

3. COMPUTATIONAL ASPECTS OF BEHAVIOURAL SYSTEM MODELS

3.1 Alternative Bases for Behavioural System Modelling

Qualitative modelling of system behaviour requires protocols and/or tools for system conceptualisation, mathematical representation, analysis and visualisation. The computational requirements are rather different from those of quantitative process-based models. However, the analyses performed are typically much less demanding, such that meaningful simulations can be implemented and executed with quite limited hardware resources.

Qualitative system models can be thought of as structural models (in the sense of Bossel, 1986) in that they provide behavioural predictions that are derived primarily from the structure of the system being studied. The underlying mathematical model can, however, be formulated in a variety of ways. In theoretical biology, there is a history of simulation based upon the identification of idealised networks of cells, genes or molecules (e.g. Kaufman, 1969, 1993; Glass, 1975). The physics and chemistry of the individual components of such systems may be complex, but important aspects of the whole system behaviour can be revealed through network-based models that formalise the interaction between large numbers of components in terms of quite simple rules. In biological systems, exchanges of information are as important as those of energy or mass (see, for example, Hopfield, 1994) and it has been found convenient to model these using Boolean functions that determine ‘on’ or ‘off’ states of system components according to varied combinations of inputs. Such analyses reveal emergent properties (such as self-organisation and the emergence of order) and afford qualitative insights into evolutionary behaviour (Kaufman, 1996). Boolean network models have found wide application in biomedicine, where the subject of study includes gene expression networks; biochemical circuits in cells; and in neural networks (e.g. Kaufman, 1974; Glass, 1975; Thomas, 1979; Fox and Hill, 2001). In computer science, the World Wide Web has also been analysed in this manner (Barabási *et al.*, 1999). The idea has been less widely applied to environmental systems, mainly in the area of climate modelling (for example, Nicolis, 1982; Saunders and Ghil, 2001).

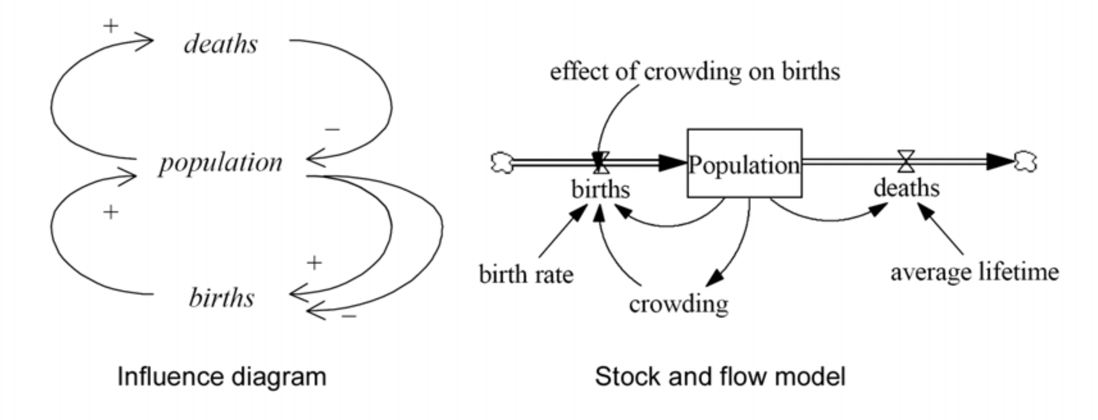
As Capobianco *et al.* (1999) show (Figure 3), it is possible to depict the geomorphological and ecological components of a coastal system in the form an interconnected network graph in which the vertices are the system components and the edges (formed by pairs of vertices) constitute the interactions between them. Karunarathna and Reeve (2005) demonstrate how network graphs for estuarine systems can be analysed to derive the dynamic behaviour of the system based largely from its topological structure. Their mathematical implementation utilises discrete Boolean logic, although Seymour (2004) has devised a slightly more complex formulation that employs continuous response functions. In both cases, simulation of network dynamics can be handled in either a synchronous or an asynchronous fashion. Synchronous models start with an initial condition and simultaneously update the values of all state variables (components) such that the system evolves over successive steps. Depending on the mathematical formulation used, these steps can be thought of either as sequential states that do not correspond directly with time or as actual time steps.

In advocating a qualitative approach to the understanding of coastal systems, Capobianco *et al.* (1999) draw upon a separate strand of research into Qualitative Reasoning (QR) that has been developed within the Artificial Intelligence (AI) community. Various forms of QR exist, although not all are readily applied to derivation of behaviour from the structural configuration of a natural (e.g. geomorphological or ecological) system. For example, the form of QR developed by Kuipers (1986, 1994) is targeted at systems than can, in principle, be defined using Ordinary Differential Equations (ODEs), but which are characterised by incomplete knowledge (such as a lack of numerical data). In this situation, ODEs are recast into Qualitative Differential Equations (QDEs) that can be solved using a specialised QR engine (such as *QSIM*; Kuipers, 1994). Qualitative calculus used in this sense is thus the qualitative analogue of the more familiar calculus of Leibniz and Newton. This methodology has been successfully used to model complex but poorly-quantified AI problems, with applications as diverse as rocket control systems, chemical reactors and robotics. The mechanistic functioning of this kind of physical system is well understood but a lack of data precludes a conventional numerical modelling approach.

The main challenge in modelling geomorphological systems often arises due to a lack of understanding at a mechanistic level rather than simply from insufficient data. However, there are alternative approaches which place more emphasis on the underlying system structure. The Qualitative Process Theory developed by Forbus (1984, 1990) is one such methodology. Another strand of literature concerns the formal analysis of system 'influence graphs' (such as that presented in Figure 2) to determine the kind of behaviour that the system is likely to exhibit. This approach has been pioneered in Operational Research, notably by Puccia and Levens (1985) and Senge (1990). Typically, qualitative analysis is used to make sense of a system prior to the construction of an equation-based simulation model (Figure 4). The latter are often based on 'stocks' and 'flows' and can be assembled using a variety of graphical user interface (GUI) driven software tools such as *Stella* and *Vensim* (refer to Section 4 of this report). Formal analysis of the influence diagram itself is possible (Smith, 2000), but such work remains in its infancy and it remains extremely difficult to devise schemes that automatically generalise the important aspects of the system behaviour (such as the identification of equilibria).

To date, applications of these various forms of modelling to geomorphological problems have been limited. A limitation of formal QR models is that, whilst it is fairly straightforward to express a geomorphological system in terms of an influence graph (Figure 3), conversion of the information that this contains into a mathematical form that can be solved by a simulation engine is more difficult. As Wolstenholme (1999) observes, despite the functionality of modern GUI-driven system simulation software, there is as yet no means to automatically convert an influence graph into a simulation model. Furthermore, the mapping and qualitative analysis of systems governed by multiple feedback circuits (or 'causal loops') actually requires considerable expertise in quantitative mathematical modelling, especially when it comes to the translation of feedback loops expressed in terms of directional influences into the flows that 'stock and flow' simulators use to represent processes. A new generation of QR-based modelling tools may remove some of these barriers: the recently released *GARP3* package (with its *HOMER* and *VISIGARP* interfaces; Bessa Marchado and Bredeweg, 2003) is more obviously suited to the investigation of geomorphological system behaviour and has already found application in aquatic ecology (Tullos *et al.*, 2004).

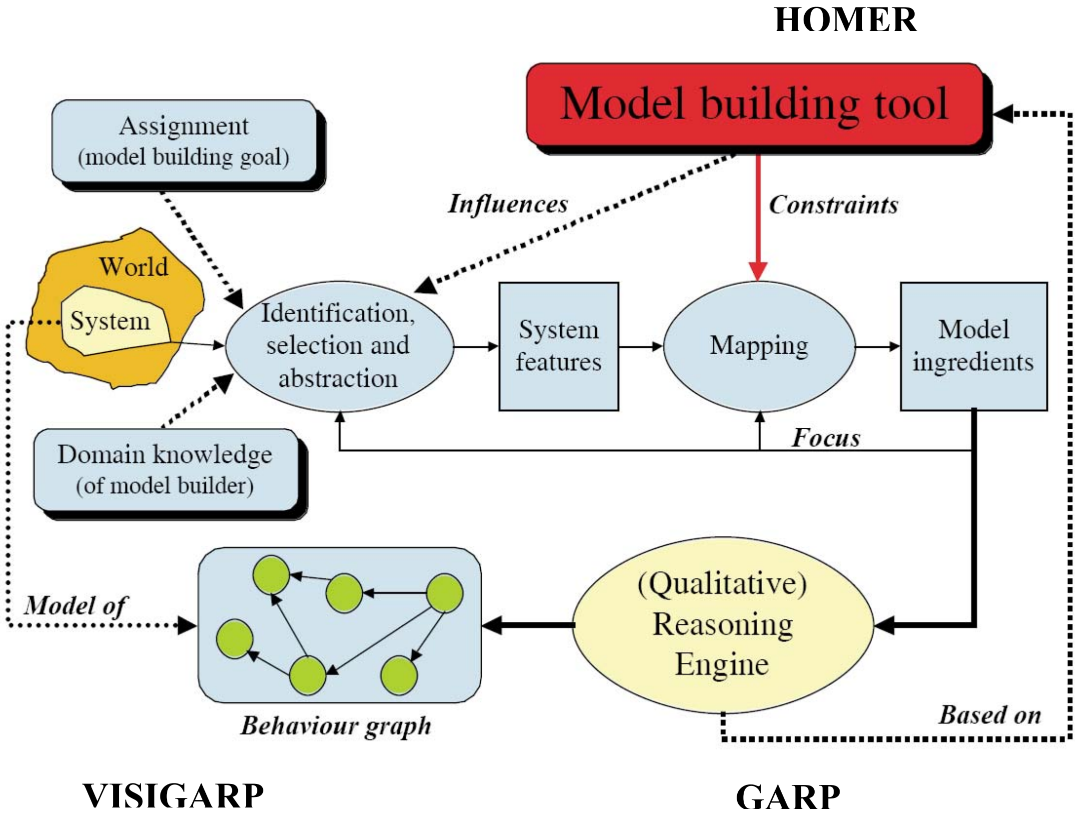
This software platform comprises closely integrated simulation engine, model development and visualisation tools (Figure 5) that allow non-specialist modellers to construct logically correct models.



(Modified from Smith, 2000)

There is, as yet, no way to fully automate the transition from influence diagram into a set of model equations (Wolstenholme, 1999).

Figure 4. Influence diagram, with causal loops, used as a prelude to 'stock and flow' simulation for a simple ecosystem model



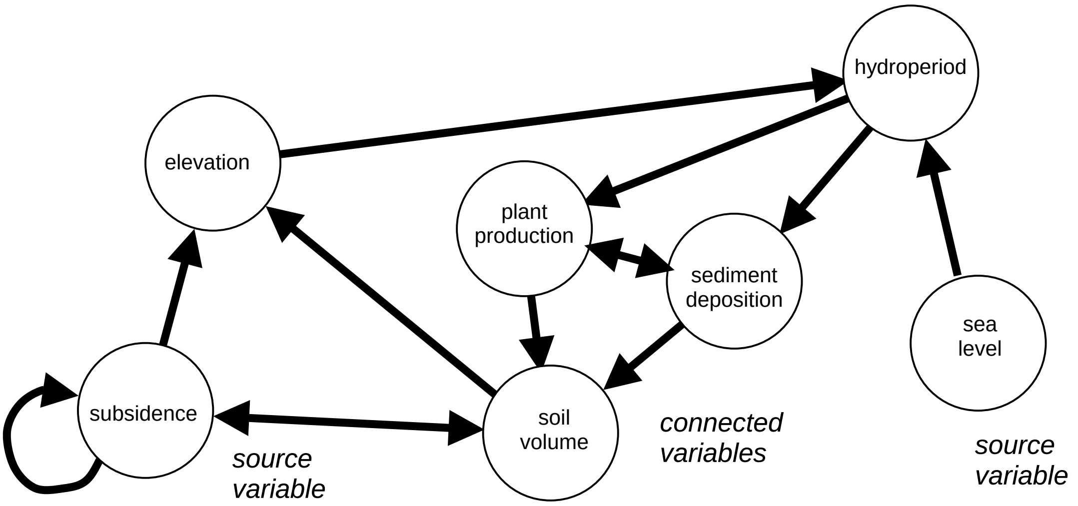
(Modified from Bessa Machado and Bredeweg, 2003)

Figure 5. Integration of simulation engine, model builder and visualisation tool using GARP, HOMER and VISIGARP qualitative modelling tools

3.2 Network-Based Models

Network-based models have unexplored potential as a means of probing the behavioural complexity of a broad class of estuarine and coastal systems. The generation of network graphs or influence diagrams is intuitive and allows the representation of both the system boundaries and the functional interactions of the internal components at a resolution that is commensurate with partial knowledge. Open system behaviour can be accommodated via the identification of external (or source) variables in addition to internal (or connected) variables (Figure 6). Also, methods already exist to automatically derive generalised aspects of system behaviour, such as the number and nature of dynamical attractors.

Boolean network models are particularly straightforward to implement. As outlined in FD2117 Project Record PR3 (Karunarathna and Reeve, 2005), a system can be defined in terms of a network of N components, which are represented by Boolean state variables. These are assigned a binary value (i.e. logical 0 or 1). The value of each state variable is determined by a Boolean function that uses logical NOT, AND, OR operators to specify the combined influence of other state variables to which it is connected. System behaviour is simulated through the by specification of a set of initial conditions (i.e. the values of all of the state variables) and the evaluation of the set of Boolean functions to arrive at a modified set of state variables. This is repeated until the system reaches an equilibrium condition.



Simple network representation of saltmarsh system that includes source variables (that influence the network dynamics but are not influenced by the network) and connected variables (that both influence and are influenced by network dynamics). Sink variables and unconnected variables can also be defined, although these are less useful for the representation of environmental systems. Self input is also possible (e.g. in the case of subsidence through autocompaction)

Figure 6. Simple network representation of saltmarsh system

Two kinds of equilibrium are routinely encountered. A steady state exists when all the state variables take the same values as their corresponding Boolean functions and no further change is possible. An oscillatory state occurs when the system falls into a cyclic sequence between two end states (possibly involving additional intermediate states).

These states can be thought of as attractors within the state space defined by all possible system states. Boolean network modelling of large biological systems ($N > 10^2$) indicate the possibility of a third 'disordered' end state, in which successive states appear non-repeating up to some arbitrary number of steps (Fox and Hill (2001) use a cut off of 1500 steps to define the boundary between periodic and disordered states). Estuary systems investigated at a meso-scale are most easily conceptualised in terms of a relatively small number of state variables: system behaviour should thus be ordered, even though both steady state and cyclical end points are possible. Strictly chaotic behaviour (in the sense of Phillips, 1992) is not possible since the set of possible system states is finite.

Simulation using a Boolean network can proceed in either of two ways. In the first, we examine the range of behaviour that is encountered across the system state variable space. The number of possible states is 2^N and exploration of the entire state space is thus only possible for small N systems. For more complex (larger N) systems, statistical sampling of the possible initial conditions is required. Studies of this kind can reveal aspects of system behaviour such as:

- The number of attractors (steady state and oscillatory equilibrium states);
- The sequence lengths of oscillatory states;
- The distribution of evolutionary sequence lengths (including maximum path length);
- The proportion of potential states that occur as viable intermediate states.

Theoretical studies of biological and other networks show that number of attractors (equilibrium states) depends on N, and also on the number of input connections between state variables, K (Kaufman, 1996; Bilke and Sjunnesson, 2001). Complexity also increases with the number of source variables (that is state variables or system components that influence the network, but are not influenced by it). Geomorphological systems would probably be small N (relative to the biological studies referred to above), but additional complexity might be expected to result from their variable and highly structured connectivity and multiple external forcing factors (e.g. sea-level, storminess, sediment supply).

The second kind of simulation involves a more intensive analysis of the evolution of the system from specific initial conditions and the interpretation of this evolutionary trajectory against the scientific understanding encapsulated in the influence diagram and Boolean functions. For a geomorphological system, model validation (in terms of the set of state variables and the formulation of the Boolean functions) could be carried out against documented sequences of historic behaviour. Also, 'what-if?' scenario modelling might be undertaken to investigate specific management questions.

It is envisaged that a prototype estuary simulation software tool based upon the Boolean network approach advocated by (Karunarathna and Reeve, 2005) will be primarily used for the second kind of analysis. Estuary users should be able to select one of the generic UK estuary system types and evaluate its behaviour under specific scenarios that reflect management questions of interest. However, as part of the proof of concept, investigation of the state variable space for different kinds of estuary is also of interest.

4. SOFTWARE PLATFORMS FOR ESTUARY BEHAVIOURAL SYSTEM MODELLING

4.1 Overview

In selecting the most appropriate software for the modelling of system behaviour an overriding criterion should be the purpose for which the modelling is being undertaken and the intended user. Once this is established, the pros and cons of particular software solutions come into play. In the context of FD2117, estuary behavioural system modelling software is needed to support the following activities (and users):

- Evaluation of the Boolean network approach formulated for FD2117 (EstSim consortium);
- Development of architecture for a Boolean network-based simulation tool (EstSim consortium; subsequent developers of tools based on this approach);
- Pilot testing of prototype simulation tool (EstSim consortium; consultants and researchers) ;
- Dissemination of research, including prototype/demonstration tool (users and consultants).

Computation of evolutionary behaviour for a simple network representation of a system can, in principle, be undertaken using readily available spreadsheet software (e.g. Microsoft *Excel* or OpenOffice *CALC*). However, for research into the properties of a range of more complex estuarine systems, or for the implementation of a user-friendly estuary management tool, spreadsheet-based software would represent a clumsy and inefficient solution. A wide range of software platforms support various kinds of system modelling. These include:

- Specialised system simulation software based on GUI-based system mapping and analysis using 'stock and flow' type models (e.g. *Simile, VisSim, Stella, ModelMaker, VenSim, Extend*), and a smaller number of solutions based upon QR (e.g. *QSIM*, *QPE*, *GARP3*).
- Numerical analysis and visualisation packages (e.g. *MATLAB, IDL, Mathematica, MathCad, Octave*).
- High-level programming languages (e.g. C, Fortran, BASIC, Java, *Adobe Flash - ActionScript*), including some that are specifically intended for system modelling applications (e.g. *Simula, PCRASTER, VisSim*).

With the objectives and target uses of EstSim in mind, relevant criteria for the selection of an appropriate software platform for the development of a Boolean network-based estuary simulator include:

- Sufficient programming capability (data types, logical and other operators, instructions and control flow, and file handling) to permit research-level use in support of proof of concept testing.
- The ability to integrate underlying model code (i.e. Boolean functions and control flow logic) with textural and/or graphic representations of system structure (e.g. behavioural statements, system diagrams).
- Visualisation capabilities.
- User interface capabilities.

- Delivery options (e.g. the ability to produce a distributable standalone software tool; support for server-based WWW delivery, or the ability to distribute model codes as ‘plug-in’ scripts for use with either proprietary or open-source application software).
- Support for open source model code and/or application software.
- Level of required expertise (ease of use and degree of training required for users).
- Cost and licensing arrangements (especially for users outside academia).

The following sections evaluate the capabilities of leading proprietary and open-source software against these criteria. The review focuses on two classes of software:

- First, packages that allow the user to construct graphical representations of the system of interest before interactively specifying the associated functional linkages and dependencies; and
- Second, that which requires the user to implement the underlying model algorithms in some form of computer code.

High-level computer programming and scripting languages are not considered further here, since these are effectively capable of implementing any software solution (though often at considerable cost in terms of development time).

4.2 Dedicated System Simulation Packages

There is now a large selection of specialised packages for systems-based simulation modelling. Many of the most sophisticated (and expensive) packages have been developed for commercial applications, notably the simulation of business organisations (up to the scale of national public sector bodies and major multi-national corporations) and industrial production processes (including real-time process control applications). Most use some form of ‘stock and flow’ representation of system quantities and processes, which is not necessarily to the most intuitive means of depicting environmental systems.

One of the pioneers in the application of systems thinking to the modelling of environmental systems was *Stella*, released by High Performance Systems, Inc (now iees systems, inc) in 1987. This object-oriented software allows the user to define the structure of the system of interest with the aid of graphical ‘drag and drop’ tools and pre-defined component types, and then to specify functional linkages interactively using libraries of mathematical operators and integration algorithms. Interactive dialogue windows also facilitate specification of boundary and initial conditions (which may be read from external data files) as well as run control information (e.g. time step and simulation duration). Models developed in this way are saved in a high-level ‘pseudo-code’ (sometimes referred to as a Modular Modelling Language, or MML), which is then interpreted into lower-level computer code when the model is run. Although this approach is computationally inefficient for the implementation of complex numerical algorithms of the kind deployed in bottom-up hydrodynamic and sediment transport models, it provides a highly intuitive platform for problem solving within a systems framework.

One applications of *Stella* has been the prototyping of coastal wetland models that can subsequently be re-coded into a high-level programming language and embedded within large scale spatial landscape simulations (Sklar *et al.*, 1994). Software of this kind allows rapid conversion of concepts to logical and mathematical expressions, while preserving a graphical representation of the linkages among variables. Combining a conceptual representation of system structure with a pseudo-code formulation of the associated mathematical algorithms in a runtime environment results in models that are particularly easy to understand and use by non-specialists.

A variety of other products now offer similar functionality and Appendix 1 summarises the main features of those that are most obviously suited to environmental modelling applications. Products vary widely in capability, ease of use and cost. The cheapest and easiest to use is probably *ModelMaker*, which offers fairly rudimentary model building and simulation capabilities, with only limited communication with other applications. At the other extreme, are heavyweight business simulation tools, such as *PowerSim* and *VisSim*, which provide sophistication modelling and application development features. *Extend* is also more obviously suited for business applications, though Odum and Odum (2000) demonstrate its potential as a tool for environmental system simulation. Given their commercial applications, such products are expensive, particularly for non-academic users. The main competitors to *Stella* in the mid-market range are probably *Simile*, which offers similar functionality at lower cost and *Vensim*, which is similar in cost but more targeted at business applications.

Comparative evaluation of these various packages in terms of their suitability for behavioural modelling of estuarine or coastal systems should clearly take account not just of their technical capabilities and cost, but also factors such as ease of use, vendor support (documentation, technical support), and user-community (including existence of established online resources and user forums). Also important, in the case of EstSim, is choice of architecture for any software tool that is developed and disseminated amongst the user community. A preliminary analysis highlights *Stella* and *Simile* as mid-range products that offer a good combination of functionality, ease of use, widespread take-up within the academic community and cost. *Stella* appears to be more widely used as an educational tool, and is strongly featured in textbooks on system modelling (e.g. Ford, 1999; Odum and Odum, 2000). However, *Simile* scores highly on its documentation, interoperability with other application software and suitability for environmental system simulation (see, for example, Mulligan and Wainwright, 2004), and is significantly less expensive. In addition, the formal logic used for model conceptualisation in *Simile* is probably more intuitive than that employed by *Stella* for users trained in geomorphology. Heavyweights such as *PowerSim* and *VisSim* probably represent overkill for the relatively simple modelling tasks envisaged here and would be expensive to roll out amongst a large and distributed user community. However, they might be a viable option for the implementation of web-based simulation tools utilising server-based computation.

Several QR-based simulation engines have been developed within the AI community, notably *QSIM* (Kuipers, 1986) and *QPE* (Forbus, 1986). However, these remain research-level tools, and require some experience in programming (both *QSIM* and *QPE* are implemented in LISP) to use effectively. Neither *QSIM* nor *QPE* is equipped with a GUI and these QR engines are not considered further here. Recently, the EU-funded NATURENET-REDIME project led by the Human Computer Studies Laboratory, University of Amsterdam (http://hcs.science.uva.nl) has developed a more tightly integrated 'workbench' for QR-based modelling that combines the *GARP* simulator, with tools for graphical model building (*HOMER*) and visualisation (*VISIGARP*). The key features of the new *GARP3* workbench have also been set out in Appendix 1. Although not suitable for network-based modelling, it does represent an important step forward in the implementation of qualitative models of environmental systems (Bessa Marchado and Bredeweg, 2003; Tullos *et al.*, 2004) and might be a viable platform for future QR-based modelling of estuary behaviour.

4.3 Numerical Analysis and Visualisation Packages

A number of sophisticated packages are available that integrate the power of a high-level programming language (such as C or Fortran) with easy to use provide tools to support algorithm development, data handling, visualisation and even the creation of standalone application software with full graphical user interface capabilities. Appendix 2 summarises the products reviewed here. Widely used commercial packages include *Mathematica*, *MATLAB*, *IDL* and *Mathcad*. Open-source offerings include *Octave*, which provides a library of functions intended to offer comparable function to *MATLAB*. All the major commercial software's offer:

- A *development environment* - including command line interface, script editor and debugger, and browsers for viewing help, workspace variables and files.
- A *library of mathematical functions* – including computational algorithms ranging from elementary functions, solvers for ordinary and partial differential equations.
- A *programming language* - possibly supporting matrix/vector operations, with control flow statements, functions, data structures and data file handling. Support for interactive 'programming' of simple tasks as well as the development of complex applications.
- A *graphics system* – including high-level functions for data visualization, image processing, animation, and presentation graphics; and low-level functions for custom graphics the design of graphical user interfaces for applications.
- An *Application Program Interface* – to permit interfacing with external Fortran or C programs.
- Optional *extensions or toolboxes* – to provide additional functionality (and function libraries) to handle specialised areas (e.g. image processing, signal processing, neural networks, or even system simulation (in the case of the *MATLAB-Simulink* module).
- Optional *compilers* – to permit the development of standalone applications that can be distributed and executed independently of the originating package. Compilation is also useful for accelerating the performance of frequently-used scripts (which run slowly in interpretive mode).

These packages are all suitable for large and complex modelling tasks and they are more than capable of handling behavioural system modelling based on a Boolean network approach.

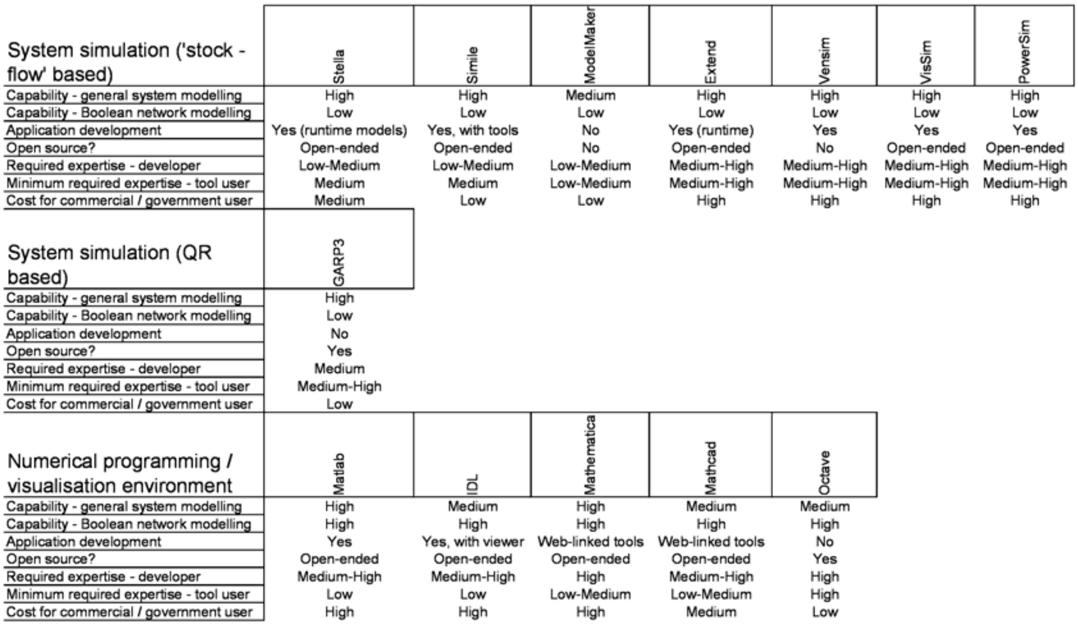
The commercial packages are well supported and have established user-communities that contribute algorithms, scripts and even complete toolboxes to online archives. *MATLAB* and *IDL* are probably most useful for reasonably efficient numerical computation of fairly large problems, whilst *Mathcad* and *Mathematica* are particularly strong on symbolic mathematics and equation solving. *Octave* is the only open-source product that falls into this category. Despite its capabilities, this must be regarded as a research tool rather than a mainstream software development product: the technical expertise required for its installation and use and its lack of application development features effectively preclude its selection as a development platform for an EstSim tool.

All are extremely powerful tools in experienced hands, but the learning curve is very high and these are not easy to use for the casual or novice user. The commercial packages are also expensive for non-academic users. Software of this kind can be classed as ‘open-ended’ since scripts are executed (and can be freely distributed and modified) as ASCII-text files. The original code can usually be ‘protected’ if desired by partially compiling into a proprietary pseudo-code. Another feature of these packages is that most are truly multi-platform (Windows, UNIX, Linux and MacOS-X).

4.4 Summary

Table 2 presents a subjective evaluation of the various software products reviewed against criteria relevant to qualitative modelling of estuary system behaviour. An important finding is that the more popular GUI-driven system simulation software such as *Stella* and *ModelMaker* are not necessarily the best options for the analysis of geomorphological systems. In particular, none of the ‘stock and flow’ based simulators are suited to network-based modelling, which is more easily accomplished using numerical analysis packages such as *MATLAB*, *Mathematica* or *Mathcad*. Even basic installations of these packages could be used to implement quite a sophisticated network-based estuary behavioural simulator without the need for additional simulation plug-ins and toolboxes.

Table 2. Subjective evaluation of selected software against criteria relevant to modelling of estuary system behaviour within FD2117



System simulation ('stock - flow' based)	Stella	Simile	ModelMaker	Extend	Vensim	VisSim	PowerSim
Capability - general system modelling	High	High	Medium	High	High	High	High
Capability - Boolean network modelling	Low	Low	Low	Low	Low	Low	Low
Application development	Yes (runtime models)	Yes, with tools	No	Yes (runtime)	Yes	Yes	Yes
Open source?	Open-ended	Open-ended	No	Open-ended	No	Open-ended	Open-ended
Required expertise - developer	Low-Medium	Low-Medium	Low-Medium	Medium-High	Medium-High	Medium-High	Medium-High
Minimum required expertise - tool user	Medium	Medium	Low-Medium	Medium-High	Medium-High	Medium-High	Medium-High
Cost for commercial / government user	Medium	Low	Low	High	High	High	High

System simulation (QR based)	GARP3
Capability - general system modelling	High
Capability - Boolean network modelling	Low
Application development	No
Open source?	Yes
Required expertise - developer	Medium
Minimum required expertise - tool user	Medium-High
Cost for commercial / government user	Low

Numerical programming / visualisation environment	Matlab	IDL	Mathematica	Mathcad	Octave
Capability - general system modelling	High	Medium	High	Medium	Medium
Capability - Boolean network modelling	High	High	High	High	High
Application development	Yes	Yes, with viewer	Web-linked tools	Web-linked tools	No
Open source?	Open-ended	Open-ended	Open-ended	Open-ended	Yes
Required expertise - developer	Medium-High	Medium-High	High	Medium-High	High
Minimum required expertise - tool user	Low	Low	Low-Medium	Low-Medium	High
Cost for commercial / government user	High	High	High	Medium	Low

5. ALTERNATIVE ARCHITECTURES FOR AN ESTSIM SOFTWARE TOOL

Implementation of a fully featured web-interfaced estuary behavioural system simulation package is beyond the immediate scope of FD2117. However, it should be possible to devise prototype software that provides valuable R&D experience (including proof of concept testing and evaluation of system conceptualisation) as well as dissemination of the behavioural systems concept and for the rationalisation of management questions that can be addressed through a systems approach.

Alternative architectures for an EstSim tool can be envisaged based around combinations of functionality and delivery. In terms of *functionality*, simulation of estuary system behaviour might take the form of interactive selection from a fixed set of scenarios interfaced to pre-computed model outcomes. The ability for the user to define custom scenarios and/or system structures would require on-demand computation and a more fully featured software tool. The highest level of functionality would also allow 'research-level' uses, such as exploration of a system's state variable space to analyse more generic aspects of its behaviour.

There is, likewise, a range of options for *delivery*. These include distribution of a standalone software application (either on CD or via the World Wide Web); implementation of a web-based decision support system; and implementation of a more sophisticated web-based decision support and analysis tool interfaced to server-side computation facilities to support a full spectrum of interactive analyses into system behaviour.

Variations on these themes are also possible including a web-based decision support system, which might be supplemented by downloadable tools and/or customisable models (either standalone or compatible with one of the leading proprietary system simulation packages). Some alternative architectures that would meet the immediate requirements of FD2117 and contribute to the enhancement of the ERP Estuary Impact Assessment System (EIAS; EMPHASYS, 2000) are summarised below.

5.1 Web-Based Estuary Simulator and Prototype Decision Support System

A model for this is ABPmer's online Estuary Guide (www.estuary-guide.net). An EstSim tool based on this architecture might comprise:

- A browser-based interface to the EstSim classification of UK estuaries, their generic system diagrams, and behavioural statements for their associated sub-systems.
- A database of pre-computed system evolution trajectories with accompanying textual descriptions, linked to a menu of estuary types and most likely management questions.
- A small database of worked examples to demonstrate the robustness of the underlying behavioural system approach when applied to test case estuaries.

This essentially constitutes a web-linked interactive database in that it has no computational capability that would support editable system diagrams or user-specified scenarios. However, it could still serve as a valuable demonstration and proof of concept tool.

5.2 A Web-Based Estuary Simulator with Server-Side Computational Capabilities

This approach would offer all of the above, with the added functionality of:

- Server-based computation of user-specific scenarios, possibly with support for customisation of estuary system diagrams to allow investigation of actual management questions for particular estuaries.

This would produce a powerful web-based application that would be a valuable addition to the existing suite of estuary modelling and management tools being developed under ERP. The simplest and least expensive approach would use a proprietary scripting language (such as Adobe's Flash) or high-level programming language to develop a custom processing engine. More complex and expensive implementations might use a heavyweight simulation package such as *PowerSim* or a *webMathematica* server. In all these cases, however, the development effort required to produce a polished product would be considerable, such that this option does not seem feasible for EstSim.

5.3 Standalone Estuary Simulator

An EstSim tool of this kind might comprise:

- A pre-compiled simulator developed using either a specialist system simulation package (e.g. *Stella*) or a compiled GUI-tool developed using a numerical computation package (e.g. *MATLAB*).
- The simulator would include pre-defined system diagrams and a set of pre-defined scenarios, but would also allow the user to specify custom system diagrams or scenarios through external (open-format) run control or steering files.
- A more sophisticated implementation might include simple graphical tools or dialogue boxes for customising system diagrams and scenarios, although this would involve considerable software development effort that is probably beyond the scope of this project.
- A supporting web site from which the tool could be downloaded as freeware, together with model files for all UK generic estuary types. Parallel distribution on CD would be possible.

This might produce a very usable product, with a simple GUI-based tool developed in (say) *MATLAB* probably offering greater ease of use than a pre-packaged *Stella*-type model (which would require users to learn how to use what is still quite a complex piece of application software). In the case of a tool built using *MATLAB* (or similar), it would be possible to maintain some of the research-level functionality for R&D into generic properties of system behaviour: this could be de-activated in a version distributed amongst a broader user community.

5.4 A Semi-Standalone Estuary Simulator

A variation on the above architecture might involve:

- A set of pre-defined model files (system diagrams, datasets and run control files) for application with a freeware or low-cost run time version of a proprietary system simulator (e.g. *Simile, Stella*).
- A supporting web site from which the model files could be downloaded as freeware, with links to freeware versions of required simulation package. Parallel distribution on CD might be possible although there are licensing issues associated with the re-distribution of third party products, even where these are available at no cost from the vendor.

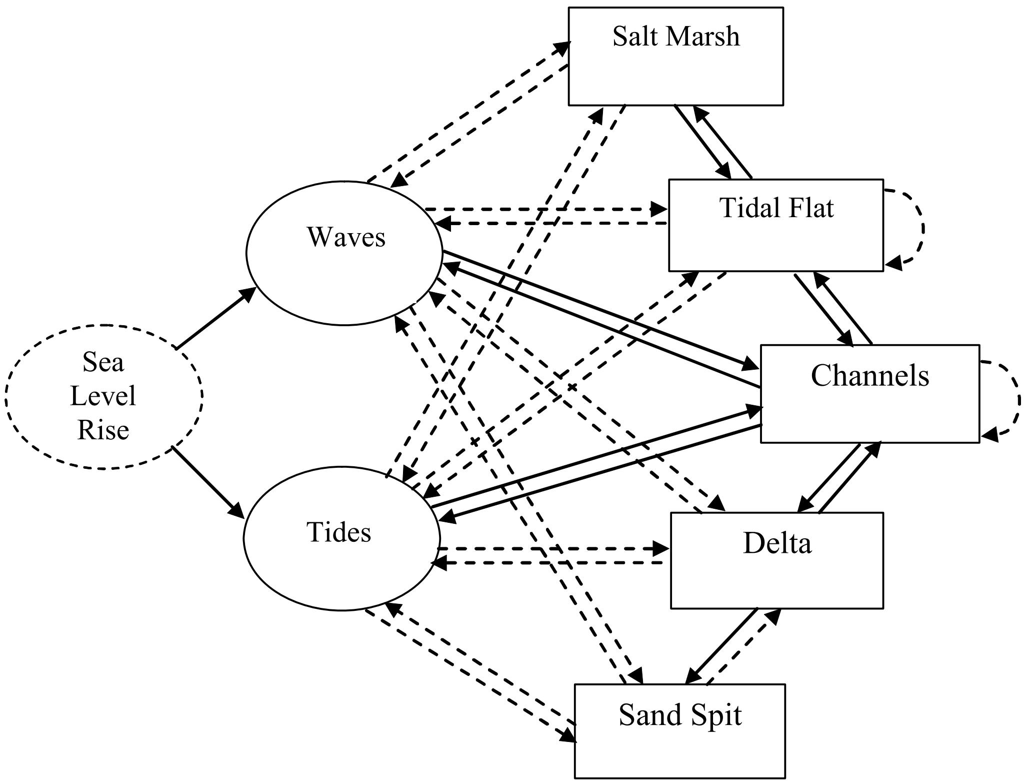
This is the least satisfactory of the above options, in that it ties the simulator to a specific proprietary package on which users must receive training.

It should be possible to implement a simulator based upon approach 1 for the purposes of FD2117. However, a more fruitful approach would probably be to combine elements of approaches 1 and 3. In the longer-term, and given sufficient resources, it might be desirable to develop a more sophisticated web-enabled product based on approach 2. This could be used to bring together different strands of qualitative modelling embracing geomorphology, hydrology, ecosystem dynamics, and socio-economics. Such a simulator could form an important component of the Estuary Management System (EMS) that is being specified under DEFRA Project FD2119.

6. BOOLEAN NETWORK ESTUARY BEHAVIOURAL SYSTEM MODEL

6.1 System Diagram Definition and Specification of Boolean Functions

Karunarathna and Reeve (2005) presented a preliminary Boolean network analysis for a simple tidal inlet comprised of seven components. Five morphological components were included (saltmarsh, tidal flat, channels, tidal delta and sand spit) and two external forcing factors (waves and tides). Their scheme is reproduced here in Figure 7.



(Taken from Karunarathna and Reeve, 2005)

Boolean and variables and functions for the above system:

Waves: W = ~sm | ~tf | cc | ~dd | ~ss;
Tides: T = ~sm | ~tf | cc | ~dd | ~ss;
Saltmarsh: SM = (~w & t & tf);
Tidal flat: TF = ((sm | cc) & t) | (~tf & ~w);
Channels: CC = (w | t) & (tf | dd) | ~cc;
Tidal delta: DD = (~w | ~t) & ~ss | (t & cc);
Sand spit: SS = (~w | ~t) & dd;

The logical operators used are AND (&) and NOT (~), and OR (|).

Figure 7. Proof of concept Boolean network model for simplified generic tidal inlet

The system diagram maps a set of influences between the morphological and process components. Whilst research in theoretical biology has focused on extremely large networks wired at random (Kaufman, 1969, 1993), the investigation of geomorphological system dynamics in this way requires that we impose a known and variable connectivity that is consistent with our a priori understanding of how the various morphological and process components interact.

A crucial assumption is that the change in a system state variable, x, in response to a set of external parameters, a, can be split into two parts: a highly non-linear term $X(x,a)$ that incorporates the interactions between system components; and a quasi-linear decay term, kx. The rate equation for x can thus be given as

$$\frac{dx}{dt} = X(x,a) - kx \qquad [1]$$

Neglecting the decay term for the time being, it is further assumed that the continuous non-linear behaviour of X can be approximated by a discontinuous Boolean variable. Taking some arbitrary threshold value of x, x_o, it is thus assumed that X becomes small for $0 < x < x_o$ but becomes appreciable for $x > x_o$ and saturates to a value of x_{max} rapidly thereafter. Below x_o, x can be considered low (or negligible) whilst above x_o, x can be considered high (or significant). These states can be represented using Boolean (logical) states of 0 and 1 respectively.

The behaviour of each system component in response to combined inputs from other components is defined using Boolean functions. Executed in synchrony at discrete time steps, the Boolean functions determine whether components (defined as Boolean variables) exist in on or off states (i.e. logical 1 or 0). The functions are assembled using conventional logical operators, AND, NOT and OR, a truth table for which is given in Table 3.

Table 3. Truth table for logical operators used to define Boolean functions

INPUT		OUTPUT
A	B	A **AND** B
0	0	0
0	1	0
1	0	0
1	1	1

INPUT		OUTPUT
A	B	A **OR** B
0	0	0
0	1	1
1	0	1
1	1	1

INPUT	OUTPUT
A	**NOT** A
0	1
1	0

A system thus defined has 2^N potential states defined by all possible combinations of the Boolean variables. For a given set of initial states for the Boolean variables, the evolutionary behaviour of the system can be traced by evaluating the corresponding Boolean functions to yield updated states for each variable. This process is repeated until the system attains one of the finite set of possible states that has been used previously (Table 4). The end to the system trajectory may take the form of a single state at which the Boolean functions evaluate to yield unaltered states for all the Boolean variables (steady state equilibrium) or a repeating sequence (cyclical equilibrium).

Table 4. Evolution of simple tidal inlet over three steps from initial state to steady end state

Step	Boolean Variables								Boolean Functions						
	w	t	sm	tf	cc	dd	ss		W	T	SM	TF	CC	DD	SS
1	0	1	1	1	1	1	1	→	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	→	1	1	0	1	1	1	0
3	1	1	0	1	1	1	0	→	1	1	0	1	1	1	0

NB. Step 3, at which Boolean functions all evaluate to give unchanged Boolean variable states. Notation is defined in Figure 7.

The Boolean functions constitute more than a concise map of the connections between a set of components. They also specify the functional behaviour of each component in response to combined inputs from other components. In essence, the behaviour of the various geomorphological components of an estuary are modelled using abstract computational devices that are akin to discrete finite automata (DFA), used in computation and electronic circuit design (Sipser, 2005).

The example presented by Karunarathna and Reeve (2005) is extremely useful as a proof of concept exercise. However, the generic estuary behavioural types presented in FD2117 Project Record PR2 '*EstSim Behavioural Statements Report* 2' (EstSim Consortium, 2004) are depicted in more detail (the generic tidal inlet, for example, comprises ten morphological components) and there are differences in the system structure (and implied functional logic). The challenge is to expand the Boolean network model to incorporate a broader set of morphological and process components and a more realistic representation of estuary behaviour at a whole system level.

Effective formalisation of geomorphological knowledge of medium-term landform dynamics within a Boolean framework requires that a number of issues be addressed. These include:

1) Conceptual basis and terminology of abstraction

Qualitative models of system dynamics are typically not specified in a way that ensures global conservation of real world quantities such as water, sediment or energy. Neither are exchanges of mass and energy the only basis for understanding the operation of a system: in biology, the widespread adoption of network-based models has been stimulated by recognition of the role played by information transmission. In geomorphology, the appropriate basis for system conceptualisation is to some extent dependent on the scale of investigation. Thus, at a micro-scale, landform morphodynamics can ultimately be reduced to a set of physical interactions. Macro-scale evolution of morphology, however, is historically and spatially contingent (e.g. Cowell and Thom, 1994; Ahnert, 1994) to a degree that currently permits only a qualitative formalisation of the interactions between the component landforms and processes. Viewed at this scale, geomorphological systems are similar to biological systems in that their structural configuration results from progression along evolutionary pathways (Hopfield, 1994), and thinking in terms of flows of information may be an equally valid approach to the conceptualisation of the outcomes of landform – process interactions. Care must still be taken to ensure that the terminology used to represent these exchanges is capable of resolving the interaction between morphology and process in a meaningful way. In the case of tides, for example, thinking merely in terms of presence or absence of tidal action is not very helpful. It might be more appropriate to think of tidal forcing in terms of high or low tidal ranges. In Boolean form, this could be handled through discrete but mutually-exclusive categories (e.g. micro-, meso- and macro-tidal ranges) that can be qualitatively associated with various estuary morphologies. These associations clearly arise through fluxes of water and sediment mediated by a variety of geomorphological processes. From a modelling perspective, however, it is the information contained within the qualitative associations that is represented rather than the physical dynamics of the underlying processes.

2) Threshold assumption of the Boolean approach

A major assumption of the Boolean model is that changes in system variables can be represented by a threshold-type function (i.e. 'on / off' or 'presence / absence'). This is not always easy to justify and a minimal binary representation is sometimes insufficient to capture the required range of geomorphological behaviour. In the case of an estuary, *increases* in external forcing (e.g. the effect of *acceleration* in the rate of sea-level rise acting through an *increase* in wave or tidal action) cannot be defined using single-bit binary logic. Equally, defining saltmarsh as a single variable precludes the expansion of saltmarsh (for example by removing flood protection from reclaimed tidal floodplain) in an estuary that already contains extensive saltmarsh. In both these cases, a variable is either present or absent (or high or low, depending on the interpretation) and if already present cannot be significantly increased. This has consequences if left unaddressed, since geomorphologists are accustomed to associating specific landform characteristics with gradients in environmental forcing. Obvious examples are the classification of barrier island morphology according to tidal range by Hayes (1979) and, at a larger scale, the estuarine facies model of Dalrymple *et al.* (1992) in which estuaries respond to broad continua of tidal, wave and fluvial energy, sediment supply and sea-level tendency. To some extent, inadequacies in the Boolean representation expose a failure to define the estuary system at a resolution commensurate with the processes to be modelled. Thus, progradational or transgressive responses of saltmarsh can be resolved through disaggregation of saltmarsh into multiple features (e.g. lower, upper and restored saltmarsh). The incorporation of gradational changes in process-related state variables such as sediment supply, sea-level rise, or tidal prism poses slightly different problems. In some cases, the creation of additional sub-system variables may be appropriate. Sediment supply, for example, might be defined in terms of a set of virtual sediment sources of incrementally increasing volume. Alternatively, certain variables might be implemented using multiple-bit logic. For example, sea-level change could be specified using two bits, the first indicating whether or not sea-level is rising and the second indicating whether any rise is at a moderate or rapid rate (giving three useful states and one that is unused). Enhancement of the Karunarathna and Reeve (2005) model along these lines appears to be needed, but must be implemented with care in order to preserve the simplicity and computational tractability of the Boolean approach.

3) Aspatial representation of spatially-distributed phenomena

The systems approach outlined in FD2117 Project Record PR2 '*EstSim Behavioural Statements Report* 2' (EstSim Consortium, 2004) generalises key aspects of estuary behaviour in an aspatial manner. Whilst features such as spits and tidal deltas are implicitly associated with the outer estuary, the distribution of other important features (tidal flat, saltmarsh) is unspecified. Most estuaries possess outer and inner zones that differ markedly in their landform assemblages and dominant processes. Indeed, a number of estuarine classifications are founded upon transitions in the relative importance of marine and fluvial processes between outer and inner estuary zones (e.g. Pritchard, 1967; Dalrymple *et al.*, 1992). Such distinctions are important for understanding the historic behaviour of estuaries in the UK. For example, the Ribble estuary in north-west England comprises an outer estuary dominated by sand flats and an inner estuary in which mudflat and saltmarsh are more extensive. Each zone has experienced distinct but rather different historic changes (van der Wal *et al.*, 2002) that would be hard to capture in an aspatial model. This is not a limitation of a Boolean network approach per se, but it does suggest the need to handle spatially distributed behaviour via the specification of separate but linked outer and inner estuary sub-systems.

6.2 Further Development of the Boolean Network Model

With the above issues in mind, further development of the Boolean network approach formulated by Karunarathna and Reeve (2005) has been undertaken using a prototype simulator. *MATLAB* was selected as our preferred development platform on account of its programming, numerical computation and visualisation capabilities, as well as the support that it provides for the subsequent development of compiled standalone tools.

The estuary system conceptualisation presented in FD2117 Project Record PR2 '*EstSim Behavioural Statements Report* 2' (EstSim Consortium, 2004) has been extended to incorporate a basic spatial division into an *outer estuary* sub-system (which interacts with elements of the adjacent *coastal system*, and potentially contains beach, spit, dune, and tidal delta or linear bank features, as well as sand or mud flat and saltmarsh) and an *inner estuary* sub-system (which is dominated by an assemblage of sub-tidal channel and intertidal flat and saltmarsh features). Such a scheme resembles the estuary facies model of Dalrymple *et al.* (1992) in that outer and inner estuary zones are likely to be dominated by marine and marine-influenced fluvial processes respectively.

The generic scheme is visualised in Figure 8, which sets out the spatial arrangement of the major morphological components and various external and internal influences on their evolutionary behaviour. We distinguish between three broad types of system component:

1) *External (imposed) forcing and interventions:*

- Ocean waves
- Longshore wave power
- Tidal range
- River discharge
- Sediment supply (marine and fluvial sand and mud)
- Wind regime (onshore winds favourable for dune formation)
- Inherited geological constraints (resistant bedrock)
- Accelerated sea-level rise
- Interventions (seawalls, erosion protection, dredging etc.)

All of the above are imposed as part of the initial conditions and are not changed by the evolutionary behaviour of the estuary. They can be altered to accommodate a change in boundary conditions, such as updrift coastal protection blocking the supply of marine sand, and specific external factors can be defined as required in order to handle anthropogenic influences such as the construction of a tidal barrage. As shown in Figure 8, strategies can optionally be defined to cover a set of interventions (e.g. 'hold the line' involves continued maintenance of all seawalls, groynes and protective structures).

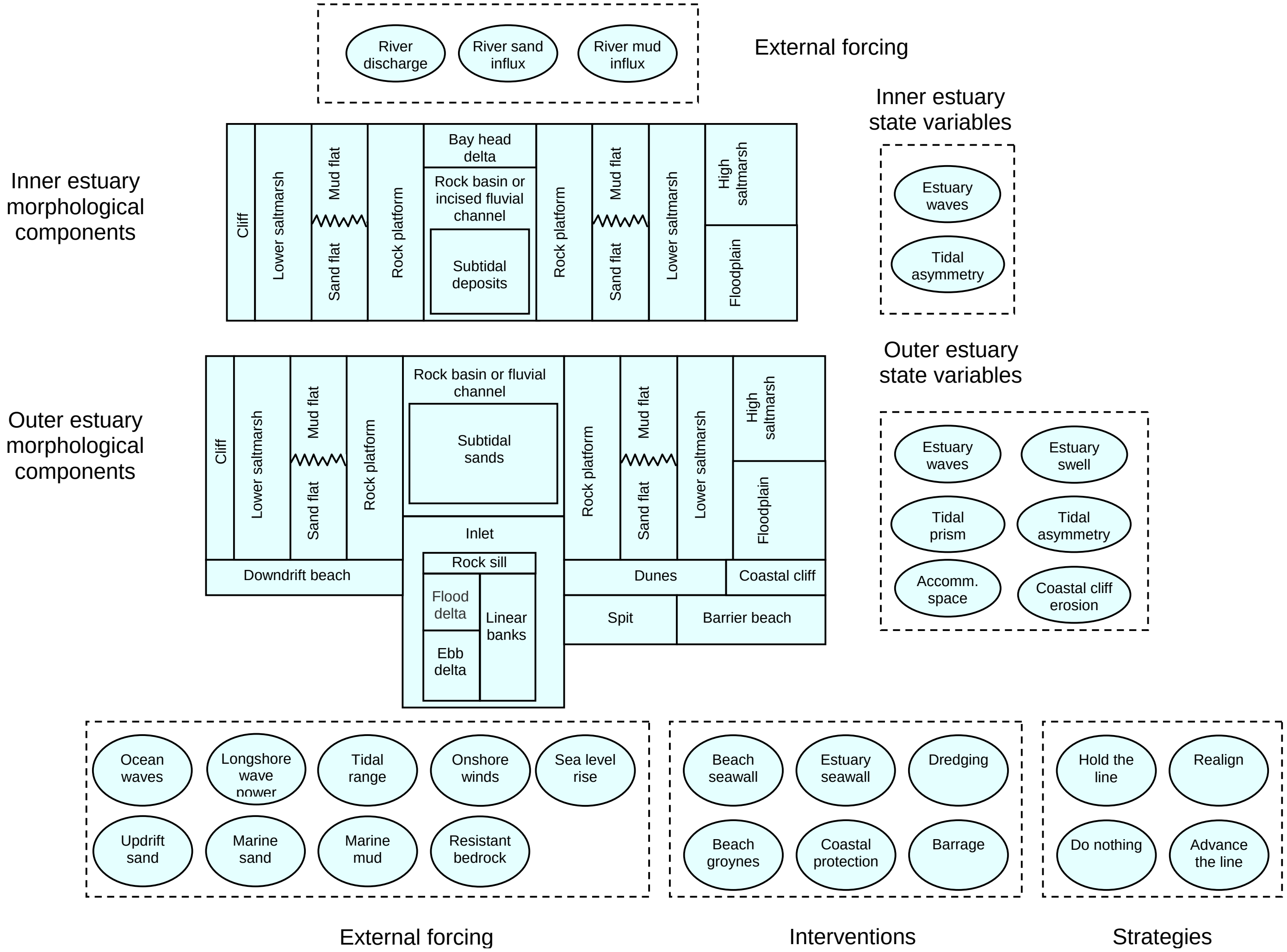


Figure 8. Morphological components of estuary simulator and various external and internal influences on their evolutionary behaviour.

2) Process state variables

State variables represent processes (*sensu lato*) that are not defined a priority but which change as estuary morphology evolves. The outer estuary is influenced by a broader range of processes, which include:

- Estuary waves, influenced mainly by prism and its control on local fetch;
- Estuary swell, influenced by ocean waves and any protection afforded by a spit;
- Tidal prism, determined by infilling of the intertidal by sedimentary landforms, especially saltmarsh within both the outer and inner estuary;
- Accommodation space, determined by infilling of both subtidal and intertidal volume;
- Tidal asymmetry, determined by intertidal area and estuary depth (a simplified representation of a Dronkers (1986) γ-type velocity asymmetry);
- Active erosion of coastal cliff, influenced by waves and coastal protection (if present).

The inner estuary is more simply represented using the following state variables:

- Estuary waves;
- Tidal asymmetry.

3) Morphological components

The basic set of morphological components is derived from FD2117 Project Record PR2 '*EstSim Behavioural Statements Report 2*' (EstSim Consortium, 2004), with a few additions to discriminate more effectively between rock-controlled and alluvial morphologies and to improve the representation of estuary-coast interaction. The basic set of components includes:

- Cliff, beach, spit and dune units (outer estuary);
- Various forms of tidal delta and inlet (outer estuary);
- Tidal sand flat and mud flat (which can co-exist in both outer and inner estuary);
- Lower and upper saltmarsh units (both outer and inner estuary);
- A rock platform that can be revealed by stripping of intertidal sedimentary cover if wave energy is high and the sediment supply negligible (both outer and inner estuary);
- Tidal floodplain that can evolve to high saltmarsh or be removed from the prism through reclamation (both outer and inner estuary);
- Estuary cliffs (in bedrock controlled situations);
- Rock basins (fjords and fjards) and fluvial channels (rias);
- Subtidal sands and muds (infilling of the estuary channel).

Intertidal depositional features are represented as vertically-stacked sedimentary units that are not mutually exclusive but which can co-exist (for example, mud flat and sand flat and low or high saltmarsh). Except in the case of rock-controlled estuaries, channels are not represented as a discrete component but are incorporated implicitly as part of a subtidal morphology that can undergo deposition and erosion depending on the availability of sediment, changes in prism and/or tidal asymmetry, and dredging. Elements of this idealised stratigraphic framework for both alluvial and rock-controlled cases are summarised schematically in Figure 9.

The main morphological components are taken to be volumetric features, some of which are mutually exclusive in a manner that is consistent with the spectrum of estuary types (e.g. different forms of inlet sand body), and some of which can co-exist (e.g. sand flat and mud flat). Morphological changes can interact directly with tidal prism as a key internal state variable; some also feedback to influence wave energy. Sediment supply variables are envisaged to take the form of stocks, or background concentrations in an essentially infinite ocean or coastal reservoir. Other forcing factors are taken to imply the existence or otherwise of sufficient tidal, wave or wind energy to accomplish the geomorphological work associated with each set of processes. Viewed in this way, the interactions or influences are more akin to information exchanges than the fluxes of energy or mass that form the basis of physically-based models.

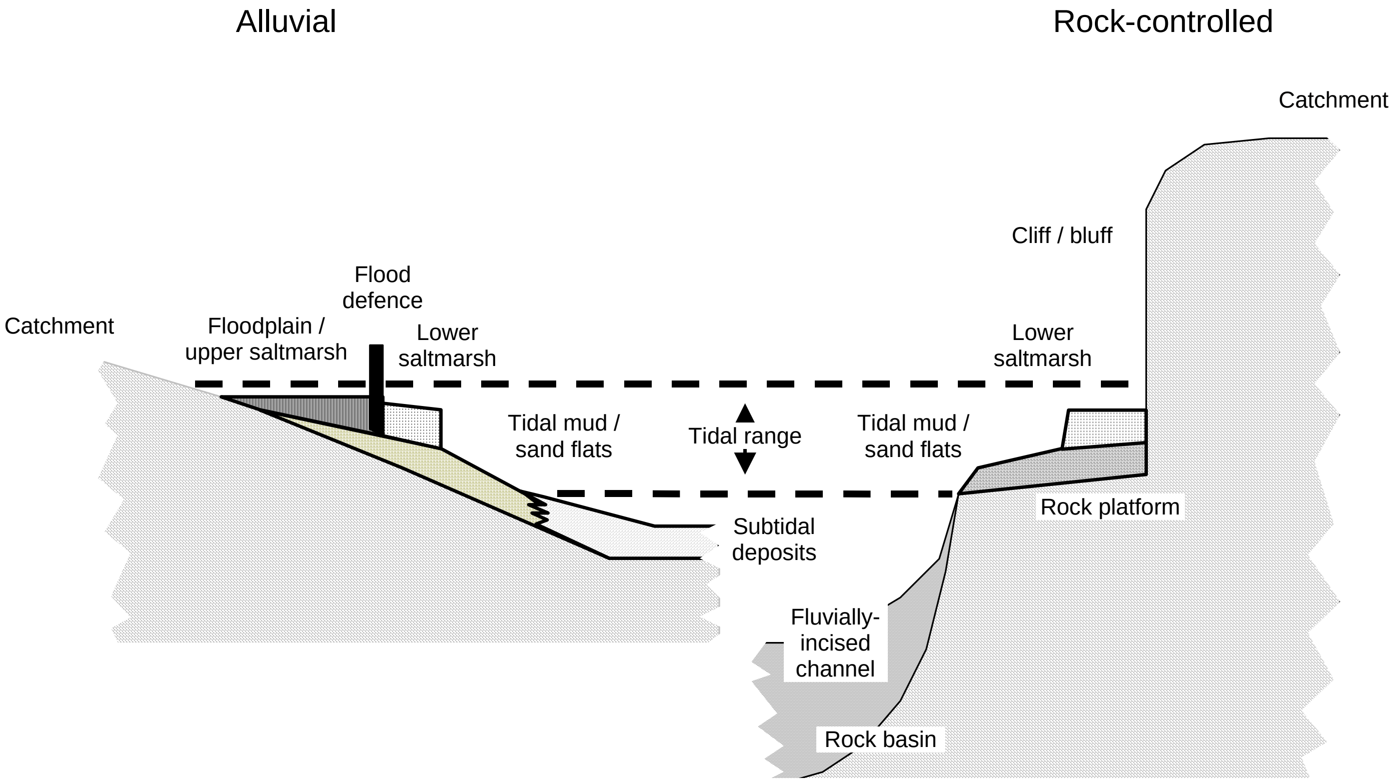
As a prelude to their implementation in model form, interactions between the major sets of system variables need to be summarised in a set of influence diagrams. Such diagrams can be highly detailed and can contain most or all of the functional logic governing system behaviour (e.g. the ‘causal loop diagrams’ employed by Puccia and Levins, 1985). In their simplest form, these diagrams merely identify the interactions and the direction of influence between all components of the system (e.g. the ‘directed graph’ of Capobianco *et al.*, 1999). Figure 10 adopts this level of detail in respect of the interaction between coastal and outer and inner estuary sub-systems. Similar influence diagrams describe the organisation of the outer and inner estuary sub-systems (Figures 11 and 12).

The influence diagrams in Figures 10-12 reflect the fact that the behaviour of the estuary system is known only in a very general sense and that each of the process and morphological variables is itself an abstraction of highly complex real world behaviour. Few of the interrelationships depicted here are well enough understood to permit the derivation of continuous mathematical functions or empirical scaling relationships. More often, what is being represented is merely accumulated experience derived variously from case studies supported by some appreciation of underlying processes studied at a small scale and in isolation.

Translating such a general level of understanding into a simulation model that is amenable both to some form of mathematical implementation and to rigorous interpretation of the modelled system behaviour is probably the major biggest challenge in qualitative modelling. In particular, there is currently no means of automatically converting an influence diagram into a simulation model (see Wolstenholme, 1999; Smith, 2000). Formulation of mathematical functions to characterise the linkages between them is likely to incur a much high ‘operator variance’ than merely sketching out the components of the system (as in Figure 8).

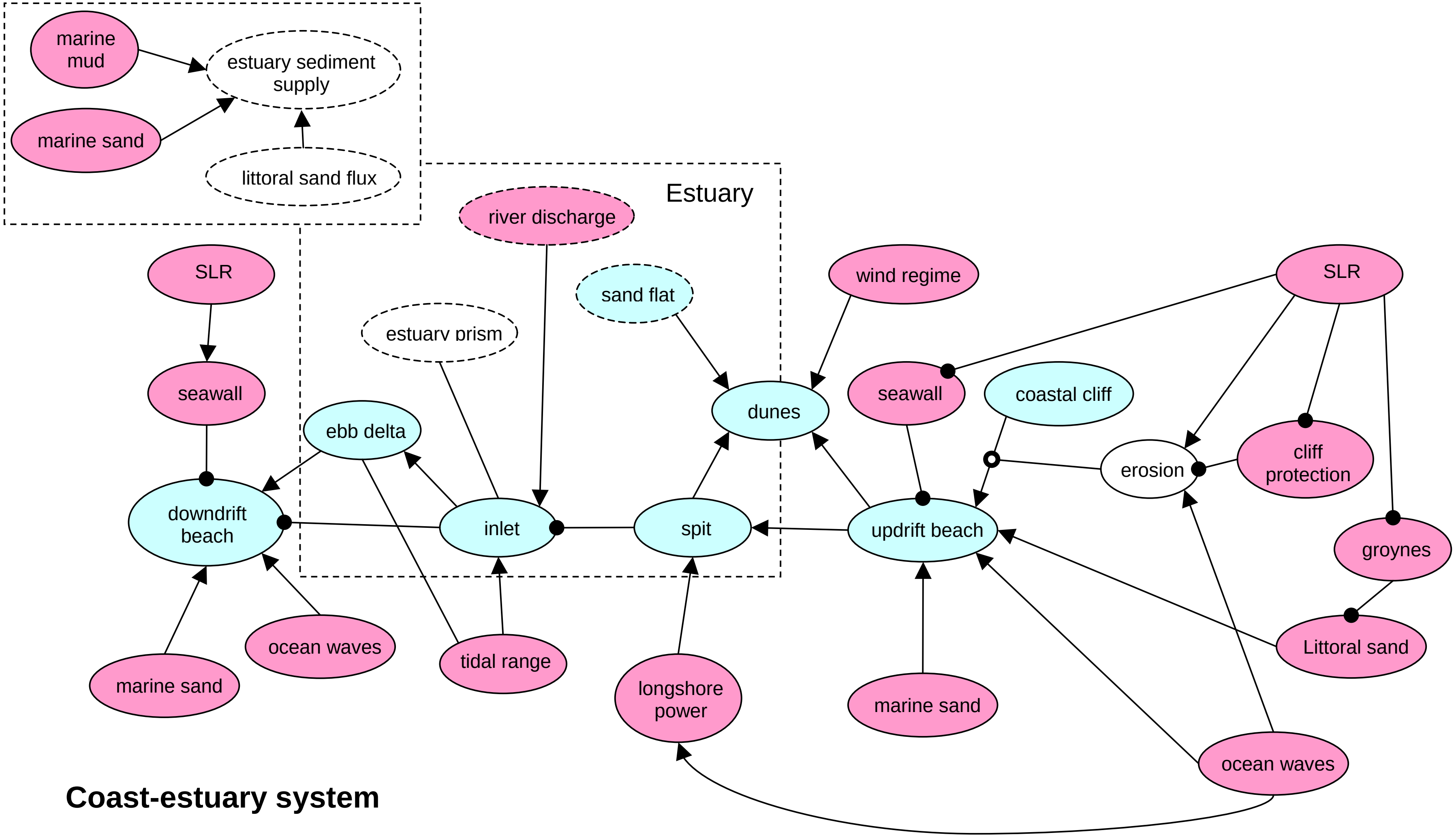
Table 5 summarises the main set of natural system variables drawn from the influence diagrams in Figures 10-12. A separate set of variables for engineering interventions is set out in Table 6. Tables 7, 8 and 9 express the linkages between these variables in the form of Boolean functions for the coast-estuary interface, outer estuary and inner estuary respectively.

System variables are indicated in lower case and are combined into associated Boolean functions, represented in upper case, defined using logical AND (& in *MATLAB*), OR (|) and NOT (~) operators. Note that self-input is also allowed, such that source variables can represent imposed external forcing and connected variables representing resistant features can exist in imposed or inherited states. Single-bit Boolean logic has been retained but model capability has been enhanced through the inclusion of additional state variables to handle gradational responses (e.g. for tidal range, where qualitative associations between micro- meso and macro-tidal conditions and estuary inlet morphology are commonly referred to in the geomorphological literature). Boolean variable states might be interpreted in terms of either 'existence' or 'tendency'. The former convention is preferred here, such that Boolean states 0 and 1 refer to negligible or significant presence (in the limit, absence or presence). This allows a more consistent handling of both process and morphological components than is possible when thinking in terms of tendency (which can, in any case, be inferred from the evolutionary behaviour of the system). In practice, such distinctions do not seem to be of critical importance provided the implementation and terminology are consistent and the user is appropriately informed.



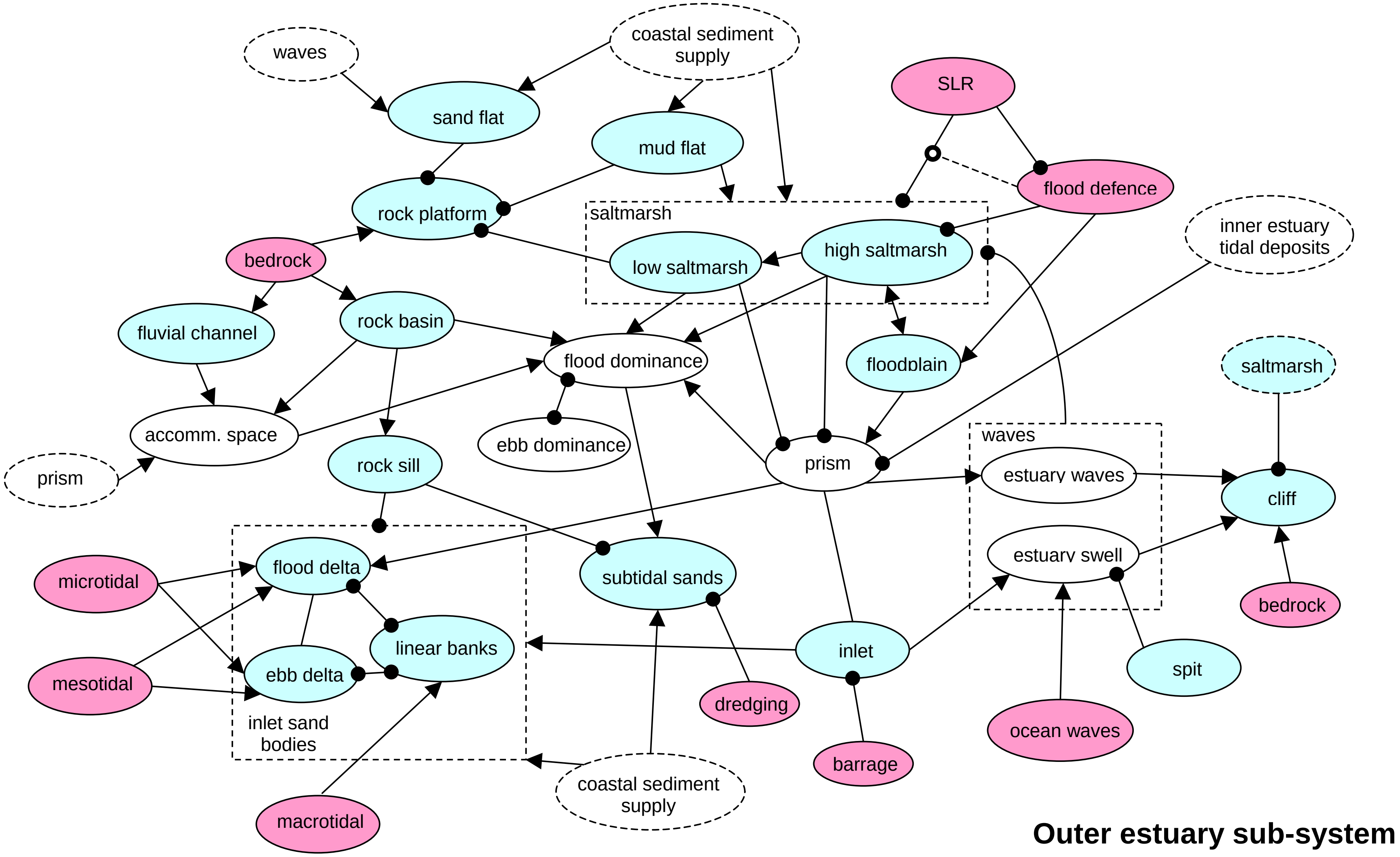
Not all morphological elements of Figure 8 are shown here.

Figure 9. Schematic illustration of idealised stratigraphic framework alluvial and rock-controlled estuary sections



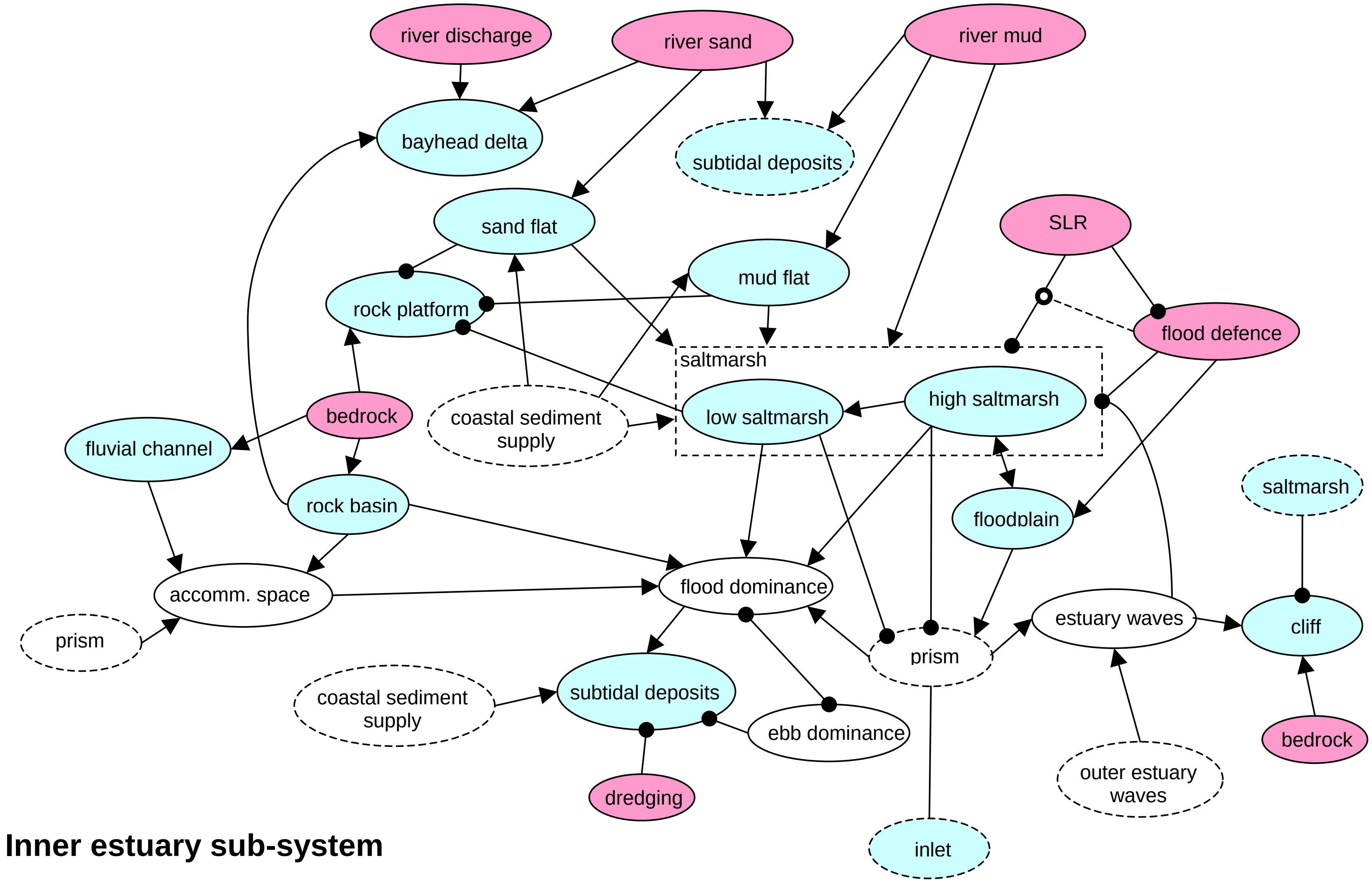
Arrows define linkages that are generally positive, whilst negative influences are indicated by —●. Dashed lines bound parts of adjacent sub-system that are not fully represented here. Shading highlights distinction between forcing, morphological and state variables. Erosion mediates transfer of material from coastal cliff to beach.

Figure 10. Influence diagram for interface between coastal and outer estuary sub-systems



Arrows indicate positive linkages, whilst negative influences are indicated by —●. Components bounded by dashed lines represent parts of adjacent sub-system that are not fully represented here. The interaction between sea-level rise (SLR) and saltmarsh is shown controlled by the presence or absence of flood defence (which effectively functions as a kind of switch).

Figure 11. Influence diagram for outer estuary sub-system



Arrows indicate positive linkages, whilst negative influences are indicated by —●.
Components bounded by dashed lines represent parts of adjacent sub-system that are not fully represented here.

Figure 12. Influence diagram for inner estuary sub-system

Table 5. Basic set of 55 Boolean variables for natural estuary systems. Coastal cliffs are here considered to be part of an imposed geology

Coastal-Estuary Sub-System	**Variable Class**		
Ocean waves			{external forcing}
Longshore wave power			{external forcing}
Littoral sand			{external forcing}
Marine (offshore) sand			{external forcing}
Marine (offshore) mud			{external forcing}
Macro tidal			{external forcing}
Meso tidal			{external forcing}
Micro tidal			{external forcing}
Sea level rise			{external forcing}
Coastal cliffs			{imposed geology}
Coastal cliff sand fraction			{imposed geology}
Coastal cliff mud fraction			{imposed geology}
Coastal cliff erosion		{state variable}	
Barrier beach - up drift	{morphology}		
Spit	{morphology}		
Inlet	{morphology}		
Barrier beach - down drift	{morphology}		
Coastal dunes	{morphology}		
Wind regime favourable for dunes			{external forcing}
Outer Estuary Sub-System			
Tidal prism		{state variable}	
Accommodation space		{state variable}	
Ebb delta	{morphology}		
Flood delta	{morphology}		
Linear banks	{morphology}		
Bedrock (resistant)			{imposed geology}
Rock sill			{imposed geology}
Rock basin			{imposed geology}
Fluvially-incised channel			{imposed geology}
Outer estuary rock platform (exposed)	{morphology}		
Outer estuary flood dominance		{state variable}	
Outer estuary ebb dominance		{state variable}	
Outer estuary subtidal deposits (incl channel sands)	{morphology}		
Outer estuary swell		{state variable}	
Outer estuary waves		{state variable}	
Outer estuary sand flat	{morphology}		
Outer estuary mud flat	{morphology}		
Outer estuary marsh - low	{morphology}		
Outer estuary marsh - high	{morphology}		
Outer estuary floodplain (0 if reclaimed)	{morphology}		
Outer estuary cliff	{morphology}		
Inner Estuary Sub-System			
Inner estuary rock platform (exposed)	{morphology}		
Inner estuary flood dominance		{state variable}	
Inner estuary ebb dominance		{state variable}	
Inner estuary subtidal deposits (channel sands/muds)	{morphology}		
Inner estuary waves		{state variable}	
Inner estuary sand flat	{morphology}		
Inner estuary mud flat	{morphology}		
Inner estuary marsh - low	{morphology}		
Inner estuary marsh - high	{morphology}		
Inner estuary floodplain (0 if reclaimed)	{morphology}		
Inner estuary cliff	{morphology}		
Bay head delta	{morphology}		
River discharge (moderate to high)			{external forcing}
River sand load (moderate to high)			{external forcing}
River mud load (moderate to high)			{external forcing}

Table 6. Additional Boolean variables to handle engineering intervention and management strategy

Coastal-Estuary Sub-System	Variable Class
Groyne field	{intervention}
Cliff toe protection	{intervention}
Seawall (coastal)	{intervention}
Barrage	{intervention}
Hold the line	{intervention - strategy}
Realign	{intervention - strategy }
Do nothing	{intervention - strategy }
Advance the line	{intervention - strategy }
Outer Estuary Sub-System	
Outer estuary flood defence	{intervention}
Outer estuary dredging	{intervention}
Inner Estuary Sub-System	
Inner estuary flood defence	{intervention}
Inner estuary dredging	{intervention}

Table 7. Boolean functions (with supporting rationale) for coast-estuary sub-system variables

```
OCEAN_WAVES = ocean_waves
```

Straightforward implementation of ocean waves as a source variable (an external forcing factor).

```
LONGSHORE_POWER = longshore_power & ocean_waves
```

```
LITTORAL_SAND = littoral_sand
MARINE_SAND = marine_sand
MARINE_MUD = marine_mud
```

```
MACROTIDAL = macrotidal & ~barrage
MESOTIDAL = (mesotidal | (macrotidal & barrage) |
      (~macrotidal & ~mesotidal & ~microtidal))
MICROTIDAL = microtidal & ~(mesotidal | macrotidal)
```

Mutually exclusive tidal range classes with range of macrotidal tidal estuary reduced by construction of a barrage.

```
SLR = slr
```

```
COASTAL_CLIFF = coastal_cliff
```

```
CLIFF_SAND = cliff_sand & coastal_cliff
CLIFF_MUD = cliff_mud & coastal_cliff
```

```
COASTAL_CLIFF_EROSION = coastal_cliff & ocean_waves & ~coastal_protection
```

```
UPDRIFT_BEACH (((littoral_sand | (coastal_cliff_erosion & cliff_sand)) & longshore_power
& ~groynes) | ((marine_sand | (coastal_cliff_erosion & cliff_sand)) & ocean_waves)) &
~(slr & seawall)
```

Beach will form/persist if there is no SLR and seawall ('coastal squeeze') and given adequate/uninterrupted sediment supply.

```
SPIT = (~spit & updrift_beach & longshore_power & (littoral_sand |
(coastal_cliff_erosion & cliff_sand)) & ~groynes) | (spit & updrift_beach &
longshore_power & (littoral_sand | (coastal_cliff_erosion & cliff_sand)) & ~(slr &
groynes))
```

A spit will form if there is a strong littoral sediment flux but will not persist if SLR impacts groyne-protected coast.

```
INLET = (~(microtidal & ~(prism | river_discharge) & spit) | mesotidal | macrotidal) &
~barrage
```

Inlet exists at high tidal range and microtidal conditions unless blocked (spit and low prism / river flow, or barrage).

```
DOWNDRIFT_BEACH = (downdrift_beach & ~(slr & seawall)) | (((ebb_delta & spit) | ~inlet)
& updrift_beach & longshore_power)
```

Forms downdrift of updrift stores (spit, beach), given longshore power and bypassing (delta). Eroded under coastal squeeze.

```
DUNES = (wind & (updrift_beach | downdrift_beach | spit) & (marine_sand | littoral_sand
| (coastal_cliff_erosion & cliff_sand) | outer_sandflat)) | (dunes & ~slr)
```

Dunes forms under favourable wind regime in association with beaches or spit and a sand source, but erode under SLR.

```
WIND = wind
```

```
GROYNES = (groynes & (~slr | hold_the_line)) | advance_the_line
```

Imposed intervention. Fail under SLR without 'hold the line' stratgey. Can also be imposed via 'advance the line' strategy.

```
COASTAL_PROTECTION = (coastal_protection & (~slr | hold_the_line)) | advance_the_line
```

```
SEAWALL = (seawall & (~slr | hold_the_line)) | advance_the_line
```

```
BARRAGE = barrage | advance_the_line
```

```
HOLD_THE_LINE = hold_the_line & ~(realign | do_nothing) | advance_the_line
REALIGN = realign & ~(hold_the_line | do_nothing | advance_the_line)
DO_NOTHING = do_nothing & ~(hold_the_line | realign | advance_the_line)
ADVANCE_THE_LINE = advance_the_line & ~(do_nothing | realign)
```

Table 8. Boolean functions (with rationale) for outer estuary sub-system variables

---

```
PRISM = (bedrock & inlet & ~(outer_marsh_low) & ~(inner_marsh_low & bayhead_delta))    |
(~bedrock & inlet & ~(outer_marsh_high | outer_flood_defence) & ~(inner_marsh_high &
(inner_marsh_low & bayhead_delta)))
```

In rock-controlled estuaries, tidal prism is reduced through formation of fringing marshes and bayhead delta. In alluvial estuaries, prism is reduced through infilling by more extensive marshes and/or removal of floodplain through reclamation. Infilling of outer estuary contributes greater reduction in prism than in inner estuary.

```
ACCOMM_SPACE = rock_basin | (fluvial_channel & ~macrotidal) | (fluvial_channel &
macrotidal & prism) | (~(rock_basin | fluvial_channel) & ((prism &
~outer_subtidal_sands) | (outer_dredging & inner_dredging)))
```

Overall estuary accommodation space is determined by presence of deep basins or fluvial channels and, according to context, the tidal prism.

```
EBB_DELTA = ~(linear_banks | rock_sill | rock_basin| fluvial_channel) & ((microtidal |
mesotidal) & inlet) & (marine_sand | (littoral_sand & longshore_power)) &
~outer_dredging
FLOOD_DELTA = ~(linear_banks | rock_sill | rock_basin | fluvial_channel) & (mesotidal |
(microtidal & spit & prism)) & (marine_sand | (littoral_sand & longshore_power)) &
~outer_dredging
LINEAR_BANKS = ~(ebb_delta | flood_delta | rock_sill | rock_basin | fluvial_channel) &
macrotidal  & (marine_sand | (littoral_sand & longshore_power))
```

Tidal deltas are mutually exclusive with respect to linear banks (which occur at higher tidal ranges). All these inlet sand bodies are restricted to alluvial estuary types (i.e. they do not occur in geologically-controlled settings).

```
BEDROCK = bedrock
```

Imposed geological constraint that discriminates between alluvial and non-alluvial estuaries.

```
ROCK_BASIN = bedrock & rock_basin
```

An inherited geological/geomorphological feature (characteristic of fjords and fjards).

```
ROCK_SILL = rock_sill & ~(ebb_delta | flood_delta | linear_banks) & rock_basin
```

An inherited geological/geomorphological feature (characteristic of fjords with overdeepened basins).

```
FLUVIAL_CHANNEL = bedrock & fluvial_channel
```

An inherited geological/geomorphological feature (characteristic of rias where a fluvially-eroded valley has been inundated).

```
OUTER_ROCK_PLATFORM = bedrock & ~(ebb_delta | flood_delta | linear_banks) &
~(outer_marsh_low | outer_sandflat | outer_mudflat)
```

Requires bedrock and a lack of sedimentary cover.

```
OUTER_FLOOD_DOMINANCE = (accomm_space | prism) & ~(outer_marsh_low & outer_marsh_high) &
~(microtidal & river_discharge) & inlet
OUTER_EBB_DOMINANCE = ~outer_flood_dominance & inlet
```

Flood-dominance occurs when depth is large (i.e. unfilled accommodation space or dredging) and when high intertidal area is not extensive. Otherwise, estuary becomes ebb-dominated.

```
OUTER_SUBTIDAL_SANDS = (~outer_subtidal_sands & ((((marine_sand | (((cliff_sand &
coastal_cliff_erosion) | littoral_sand) & longshore_power) ) & ~ rock_sill) &
outer_flood_dominance & (flood_delta | linear_banks))) & ~outer_dredging) |
(outer_subtidal_sands & ~outer_dredging)
```

Infilling of estuary subtidal occurs given coastal sediment supply, inlet sand bodies and flood dominance (but not if rock sill is present, or if material is removed through dredging).

```
OUTER_ESTUARY_SWELL = ocean_waves & inlet & ~spit
```

```
OUTER_ESTUARYWAVES = prism
```

Propagation of any ocean waves is facilitated by inlet but impeded by spit. Waves are generated within the outer estuary itself if the fetch is large enough - approximated here by the tidal prism.

```
OUTER_SANDFLAT = ((marine_sand | (((cliff_sand & coastal_cliff_erosion) | littoral_sand)
& longshore_power)) & ~ rock_sill) & (outer_estuary_swell | outer_estuarywaves |
macrotidal)
```

Forms under conditions of sand supply (from one or more of marine, coastal cliff or beach sources, unimpeded by rock sill) and significant wave and/or macrotidal action.

Table 8. (Continued)

OUTER_MUDFLAT = (~outer_mudflat & (marine_mud | (cliff_mud & coastal_cliff_erosion)) & ~(outer_estuary_swell & outer_estuarywaves)) | ((marine_mud | (cliff_mud & coastal_cliff_erosion)) & ~(outer_estuary_swell & outer_estuarywaves))
Forms under conditions of mud supply (from marine or coastal cliff sources) and negligible wave action.

OUTER_MARSH_LOW = (~outer_marsh_low & (outer_mudflat | outer_sandflat) & (outer_flood_defence | outer_marsh_high | outer_cliff) & (marine_mud | (cliff_mud & coastal_cliff_erosion)) & ~(outer_estuary_swell & outer_estuarywaves)) | (outer_marsh_low & (outer_mudflat | outer_sandflat) & ~(outer_estuary_swell) & ~(((outer_flood_defence | outer_marsh_high | bedrock) & slr) & (slr | ~(marine_mud | (cliff_mud & coastal_cliff_erosion)))))
Forms under conditions of mud supply (from marine or coastal cliff sources) and negligible wave action. Can be lost through coastal squeeze if backed by higher marsh, cliff, or flood defence and subjected to SLR.

OUTER_MARSH_HIGH = (~outer_marsh_high & (marine_mud | (cliff_mud & coastal_cliff_erosion)) & ~bedrock & ~outer_flood_defence & ~(outer_estuary_swell & outer_estuarywaves)) | (outer_marsh_high & ~outer_flood_defence & ~(slr & ~(marine_mud | (cliff_mud & coastal_cliff_erosion))) & ~outer_estuarywaves)
Forms under conditions of mud supply (from marine or coastal cliff sources) and negligible wave action. Can be lost if subjected to SLR with inadequate sediment supply.

OUTER_FLOOD_DEFENCE = ((outer_flood_defence & (~slr | hold_the_line)) | ~outer_flood_defence & advance_the_line) & ~realign
Intervention that protects flood plain from tidal inundation. Will fail under SLR unless maintained ('hold the line'). Can be removed or constructed through broader 'realignment' and 'advance the line' strategies respectively.

OUTER_FLOODPLAIN = ~(outer_cliff & bedrock) & outer_flood_defence
Floodplain in this sense refers to reclaimed upper estuarine intertidal that can potentially be converted to saltmarsh (through removal of the flood defence). Assumed not to be a significant component in rock-controlled estuaries.

OUTER_CLIFF = bedrock & ~(outer_marsh_high | outer_marsh_low | inner_flood_defence | inner_floodplain) & (outer_estuarywaves | outer_estuary_swell)
Estuary cliff characteristic of rock-controlled settings, with wave action and no fronting saltmarsh.

OUTER_DREDGING = outer_dredging

Table 9. Boolean functions (with rationale) for inner estuary sub-system variables

---

```
INNER_ROCKPLATFORM = bedrock & ~(inner_marsh_low | inner_sandflat | inner_mudflat)
```

```
INNER_FLOOD_DOMINANCE = (accomm_space & prism) & ~(inner_marsh_low & inner_marsh_high) &
~(inner_marsh_low & inner_dredging) & ~((microtidal | mesotidal) & river_discharge) &
inlet
```

```
INNER_EBB_DOMINANCE = ~inner_flood_dominance & inlet
```

```
INNER_SUBTIDAL_DEPOSITS = (((outer_subtidal_sands & inner_flood_dominance) | river_sand
| river_mud) & ~inner_dredging & ~(rock_basin & rock_sill)) |  (inner_subtidal_deposits
& ~inner_dredging)
```

```
INNER_ESTUARYWAVES = outer_estuarywaves & ~(inner_marsh_high | inner_marsh_low |
inner_sandflat | inner_mudflat)
```

Waves can propagate from high energy outer estuary, or else be generated within the inner estuary itself if the fetch is large enough.

```
INNER_SANDFLAT = inner_subtidal_deposits & ((((marine_sand | (((cliff_sand &
coastal_cliff_erosion) | littoral_sand) & longshore_power)) & ~ rock_sill) &
outer_flood_dominance) | river_sand) | (inner_sandflat & ~(slr & ~((((marine_sand |
(((cliff_sand & coastal_cliff_erosion) | littoral_sand) & longshore_power)) & ~
rock_sill) & outer_flood_dominance) | river_sand)))
```

```
INNER_MUDFLAT = (~inner_mudflat & ((marine_mud | (cliff_mud & coastal_cliff_erosion)) |
river_mud) & ~inner_estuarywaves) | (((marine_mud | (cliff_mud & coastal_cliff_erosion))
| river_mud) & ~(inner_estuarywaves & inner_ebb_dominance & outer_ebb_dominance & slr))
```

```
INNER_MARSH_LOW = (~inner_marsh_low & (inner_mudflat | inner_sandflat) &
(inner_flood_defence | inner_marsh_high | inner_cliff) & (((marine_mud | (cliff_mud &
coastal_cliff_erosion)) | river_mud) & ~(inner_estuarywaves | (slr &
(inner_flood_defence | inner_cliff))))) |  (inner_marsh_low & (inner_mudflat |
inner_sandflat) & (~(((inner_flood_defence | inner_marsh_high | inner_cliff | bedrock) &
slr) | (slr | ~((marine_mud | (cliff_mud & coastal_cliff_erosion)) | river_mud)))))
```

```
INNER_MARSH_HIGH = (~inner_marsh_high & ((marine_mud | (cliff_mud &
coastal_cliff_erosion)) | river_mud) & ~inner_flood_defence & ~bedrock & (inner_mudflat
| inner_sandflat) & ~inner_estuarywaves) |  (inner_marsh_high & ~inner_flood_defence &
~(slr & ~((marine_mud | (cliff_mud & coastal_cliff_erosion)) | river_mud) &
inner_estuarywaves))
```

```
INNER_FLOOD_DEFENCE = (inner_flood_defence & (~slr | hold_the_line)) |
~inner_flood_defence & advance_the_line & ~realign
```

```
INNER_FLOODPLAIN = ~(inner_cliff & bedrock) & inner_flood_defence
```

```
INNER_CLIFF = bedrock & ~(inner_marsh_high | inner_flood_defence | inner_floodplain |
inner_marsh_low)
```

Estuary cliff characteristic of rock-controlled settings where there is no fronting saltmarsh.

```
INNER_DREDGING = inner_dredging
```

```
BAYHEAD_DELTA = rock_basin & river_discharge & river_sand & ~((bayhead_delta &
inner_flood_defence & slr & ~(river_discharge | river_sand)))
```

Sand and/or gravel deposits at estuary head under conditions of significant river discharge and coarse sediment load. Less accommodation space is available in rock controlled estuaries, and can be replaced by saltmarsh in either rock-controlled or alluvial settings. Delta can be lost under SLR if sediment supply is removed.

```
RIVER_DISCHARGE = river_discharge
RIVER_SAND = river_sand & river_discharge
RIVER_MUD = river_mud & river_discharge
```

External fluvial system inputs

The operation of the Boolean logic can be illustrated more clearly with reference to the truth table for one of the simpler functions. As an example, Figure 13 shows the logic for coastal cliff erosion. Inspection of the truth table and its associated binary decision tree confirms that only one combination of input values (presence of the cliff morphological element, existence of significant wave action, and absence of coastal protection) results in the Boolean expression evaluating to TRUE (i.e. a logical 1). All other combinations yield FALSE (logical 0).

Qualititative reasoning. Our representation of coastal cliff erosion dictates that this requires the presence of both a cliff and significant wave action but that erosion will cease with installation of cliff toe protection.

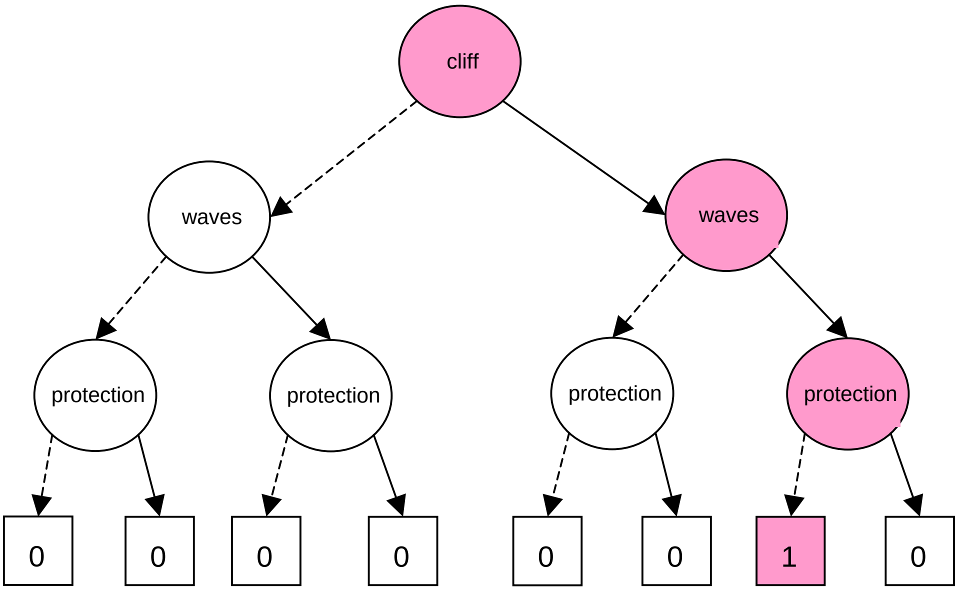
Boolean function that incorporates this reasoning (taken from Table 7):

```
COASTAL_CLIFF_EROSION = coastal_cliff & ocean_waves & ~coastal_protection
```

Truth table for this function:

COASTAL_CLIFF_EROSION	coastal_cliff	ocean_waves	coastal_protection
FALSE	0	0	0
FALSE	0	0	1
FALSE	0	1	0
FALSE	0	1	1
FALSE	1	0	0
FALSE	1	0	0
FALSE	1	0	1
TRUE	1	1	0
FALSE	1	1	1

Binary decision tree:



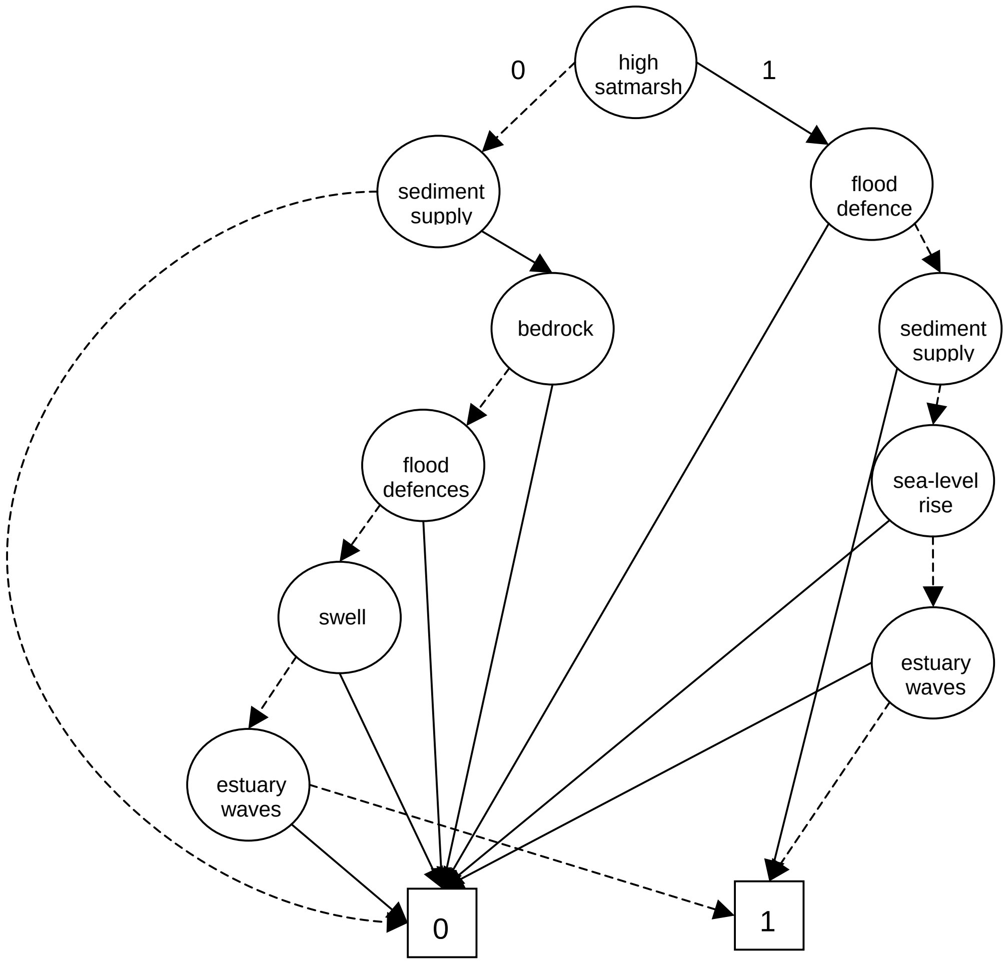
In binary decision tree, solid 'edges' lead to high (true) states and dashed 'edges' lead to low (false) states.

Figure 13. Illustration of Boolean logic for coastal cliff erosion

Although the formalisation of knowledge represented by Figure 13 is straightforward, the truth tables for more complex functions can become very large. The function for the high saltmarsh element within the outer estuary (Table 8), for example, includes 9 terms (including self input). The truth table thus contains 512 (i.e. 2^9) states. To some extent, this additional complexity can be managed by the aggregation of related terms into temporary variables. Thus, a 'sediment supply' variable can be created from the combination of marine mud, cliff mud and coastal cliff erosion, thereby reducing the function to 7 terms with 128 potential states. Such aggregate variables can be eliminated once the functions are fully coded and tested but could equally be retained in the final system representation if less importance is attached to minimising the eventual variable set.

Another way of managing function complexity exploits the fact that Boolean functions are often extremely sparse and contain many zero states and outcomes that depend on a small fraction of the possible 2^N states. In this case, the Boolean decision trees can be implemented more efficiently by judicious ordering of the variables, deletion of redundant nodes and sharing of equivalent sub-graphs to give a Binary Decision Diagram (BDD; Bryant, 1986; Minato and Arimura, 2004). Figure 14 shows a BDD for the high saltmarsh element of the outer estuary. A new saltmarsh forms in the presence of a sediment supply (here, a function of marine mud and coastal cliff-derived mud) in settings not characterised by bedrock, enclosure of the upper intertidal by flood defences, and significant swell or estuary-generated waves. Existing saltmarsh will persist if it remains unenclosed by flood defences and if sustained by sediment (in which case it can track sea-level rise) or in the absence of sea-level rise and estuary waves.

Although computational tools are available to support rigorous graph-based analysis of Boolean functions (e.g. Bryant, 1986), the construction of influence diagrams for geomorphological systems and the formalisation of geomorphological knowledge into sets of Boolean functions remains much more subjective. It follows that neither the influence diagrams nor the derived Boolean functions presented here are unique. Given a set of morphological components and process variables (Figure 8) different individuals (or 'experts') will inevitably conceptualise and code system behaviour in terms of different indicative linkages and Boolean functions. Ideally, the variance associated with this procedure should be investigated, and optimal representations arrived at through consensus (e.g. via at expert workshop). The functions set out in Tables 7-9 should thus be considered purely as a starting point from which to explore the potential and limitations of a Boolean approach. They do *not* constitute a definitive formalisation of coastal and estuary geomorphological knowledge, and both the functions and the variable set is likely to require further customisation for application to specific estuaries.



See text for explanation

Figure 14. Binary decision diagram for outer estuary high saltmarsh.

6.3 Initial Simulation of Generic Estuary Types

The first test of the EstSim model is whether or not it is capable of discriminating between the seven generic estuary types defined in FD2117 Project Record PR2 '*EstSim Behavioural Statements Report* 2' (EstSim Consortium, 2004). These are defined at a fairly high level of abstraction (see Appendix 3 for a summary of the main process and morphological components) and some further effort is needed to implement each type using the expanded variable set developed in the preceding section.

From the 67 possible variables in Tables 5 and 6 a minimal set of 50 can be used to define all of the generic estuary types in their natural state (neglecting accelerated sea-level rise). A matrix of variables and their initial states for each estuary type is given in Table 10. Only forcing variables (including imposed geological features) are defined at the outset for each type: morphological components and state variables are otherwise free to evolve. Not all possibilities are considered for each type. Fjords, fjards and rias, for example, are modelled without the potential for spit formation. By default, a mesotidal range is assumed, except for funnel-shaped estuaries and embayments, which are here assumed to be macrotidal.

Table 10. Initial conditions used for simulation of generic estuary types, using minimal variable set for natural system (i.e. no interventions or accelerated sea-level rise modelled at this stage)

Estuary Type:	Fjord	Fjard	Ria	Spit-Enclosed	Funnel	Embayment	Inlet
ocean_waves	1	1	1	1	1	1	1
longshore_power	0	0	0	1	0	0	1
littoral_sand	0	0	0	1	1	1	1
marine_sand	0	0	1	1	1	1	1
marine_mud	0	0	1	0	1	1	1
macrotidal	0	0	0	0	1	1	0
mesotidal	1	1	1	1	0	0	1
microtidal	0	0	0	0	0	0	0
updrift_beach	0	0	0	0	0	0	0
spit	0	0	0	0	0	0	0
inlet	1	1	1	1	1	1	1
rock_basin	1	1	0	0	0	0	0
fluvial_channel	0	0	1	0	0	0	0
downdrift_beach	0	0	0	0	0	0	0
dunes	0	0	0	0	0	0	0
wind	1	1	1	1	1	1	1
prism	1	1	1	1	1	1	1
accomm_space	1	1	1	1	1	1	1
ebb_delta	0	0	0	0	0	0	0
flood_delta	0	0	0	0	0	0	0
linear_banks	0	0	0	0	0	0	0
bedrock	1	1	1	0	0	0	0
rock_sill	1	0	0	0	0	0	0
outer_rock_platform	0	0	0	0	0	0	0
outer_flood_dominance	0	0	0	0	0	0	0
outer_ebb_dominance	0	0	0	0	0	0	0
outer_subtidal_sands	0	0	0	0	0	0	0
outer_estuary_swell	0	0	0	0	0	0	0
outer_estuarywaves	0	0	0	0	0	0	0
outer_sandflat	0	0	0	0	0	0	0
outer_mudflat	0	0	0	0	0	0	0
outer_marsh_low	0	0	0	0	0	0	0
outer_marsh_high	0	0	0	0	0	0	0
outer_floodplain	0	0	0	0	0	0	0
outer_cliff	0	0	0	0	0	0	0
inner_rockplatform	0	0	0	0	0	0	0
inner_flood_dominance	0	0	0	0	0	0	0
inner_ebb_dominance	0	0	0	0	0	0	0
inner_subtidal_deposits	0	0	0	0	0	0	0
inner_estuarywaves	0	0	0	0	0	0	0
inner_sandflat	0	0	0	0	0	0	0
inner_mudflat	0	0	0	0	0	0	0
inner_marsh_low	0	0	0	0	0	0	0
inner_marsh_high	0	0	0	0	0	0	0
inner_floodplain	0	0	0	0	0	0	0
inner_cliff	0	0	0	0	0	0	0
bayhead_delta	0	0	0	0	0	0	0
river_discharge	1	1	1	1	1	0	0
river_sand	1	1	1	1	1	0	0
river_mud	0	1	1	1	1	0	0

Simulated evolutionary end points for each set of generic initial conditions are presented in Table 11. The evolutionary behaviour for each of the 7 types is briefly described below.

Type 1: Fjord. This is modelled as a sediment deficient system, with the exception of a supply of river sand (which allows formation of a bay head delta). There are no tidal deltas or linear banks; no intertidal sedimentary units; and tidal prism and accommodation remain large (i.e. unfilled by sediment). A steady state, characterised by active cliffs and rock platforms emerges after just 3 steps. Some fjords do have spits. The Boolean functions in Table 7 allow spits to form where an updrift beach is fed by littoral sand flux, but allow the presence of a rock sill to impede coarse sediment transport into the fjord. Some assumption of this kind is required in order to prevent infilling of the accommodation space and conversion to another estuary type (such as a funnel-shaped estuary). Alternatively, a spit can originate from reworking of relict sediment sources (e.g. offshore fluvial-glacial sands and gravels). In this case, a spit might persist where an updrift barrier beach is imposed, longshore transport is significant and sea-level rise is low. Such a feature would degenerate under accelerated sea-level rise (i.e. with no new sediment input).

Type 2: Fjard. As noted in the EstSim behavioural statements, the distinction between fjords and fjards is rather subtle. The former are over-deepened through glacial erosion and exhibit a shallower sill near the mouth; the latter typically contain rather more extensive intertidal deposits, including inner estuary saltmarsh. Here, the presence or absence of an imposed rock sill allows slightly different behaviours and a supply of mud (fluvial or marine) allows the formation of some saltmarsh in the limited accommodation space. With only a river sand influx, a steady state emerges after 4 steps (not shown in Table 11 but similar to a fjord, with the addition of inner estuary sand flats). The addition of a marine mud input (as specified in the initial conditions of Table 10) leads to an evolution towards a steady state after 5 steps in which saltmarsh forms within the inner estuary.

Type 3: Ria. Rias can potentially contain a range of intertidal depositional features, which implies a supply of sand and/or mud. However, the accommodation space remains substantially unfilled by Holocene sedimentation and their inherited fluvial planform contrasts with that of sediment-rich funnel-shaped (type 5) systems. This can be accommodated via the inclusion of a fluvially-incised outer estuary channel. This effectively imposes an accommodation space that does not favour wave-driven movement of coarse sediment into the estuary (wave action less effective in larger water depth) and impedes bypassing of sediment across the inlet given relatively deep water and an absence of tidal deltas. Deposition in the inner estuary is facilitated by marine fluvial mud input. Dunes can form if a spit is present. With the initial configuration given in Table 10, a cyclical endpoint (involving inner estuary marsh) is reached at steps 4 – 6.

Table 11. End states simulated for generic estuary types using initial conditions specified in Table 10

Estuary Type:	Fjord	Fjard	Ria	Spit-Enclosed	Funnel	Embayment	Inlet
ocean_waves	1	1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
longshore_power	0	0	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
littoral_sand	0	0	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
marine_sand	0	0	0 ↔ 0	1 ↔ 1	1 ↔ 1	1 ↔ 1	0 ↔ 0
marine_mud	0	0	0 ↔ 0	1 ↔ 1	1 ↔ 1	1 ↔ 1	0 ↔ 0
macrotidal	0	0	0 ↔ 0	0 ↔ 0	1 ↔ 1	1 ↔ 1	0 ↔ 0
mesotidal	1	1	1 ↔ 1	1 ↔ 1	0 ↔ 0	0 ↔ 0	1 ↔ 1
microtidal	0	0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
updrift_beach	0	0	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
spit	0	0	1 ↔ 1	1 ↔ 1	0 ↔ 0	0 ↔ 0	1 ↔ 1
inlet	1	1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
rock_basin	1	1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
fluvial_channel	0	0	1 ↔ 1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
downdrift_beach	0	0	0 ↔ 0	1 ↔ 1	0 ↔ 0	0 ↔ 0	1 ↔ 1
dunes	0	0	1 ↔1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
wind	0	0	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
prism	1	0	1 ↔ 1	**0 ↔ 0**	**0 ↔ 0**	**0 ↔ 0**	**0 ↔ 0**
accomm_space	1	1	1 ↔ 1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
ebb_delta	0	0	0 ↔ 0	1 ↔ 1	0 ↔ 0	0 ↔ 0	1 ↔ 1
flood_delta	0	0	0 ↔ 0	1 ↔ 1	0 ↔ 0	0 ↔ 0	1 ↔ 1
linear_banks	0	0	0 ↔ 0	0 ↔ 0	1 ↔ 1	1 ↔ 1	0 ↔ 0
bedrock	1	1	1 ↔ 1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
rock_sill	1	1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
outer_rock_platform	1	1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
outer_flood_dominance	1	1	1 ↔ 1	0 ↔ 0	**1 ↔ 1**	**1 ↔ 1**	0 ↔ 0
outer_ebb_dominance	0	0	0 ↔ 0	1 ↔ 1	**0 ↔ 0**	**0 ↔ 0**	1 ↔ 1
outer_subtidal_sands	0	0	0 ↔ 0	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
outer_estuary_swell	1	1	0 ↔ 0	0 ↔ 0	1 ↔ 1	1 ↔ 1	0 ↔ 0
outer_estuarywaves	1	1	1 ↔ 1	**1 ↔ 1**	**1 ↔ 1**	**1 ↔ 1**	**1 ↔ 1**
outer_sandflat	0	0	1 ↔ 1	**0 ↔ 0**	1 ↔ 1	1 ↔ 1	**1 ↔ 1**
outer_mudflat	0	0	0 ↔ 0	1 ↔ 1	**0 ↔ 0**	**0 ↔ 0**	1 ↔ 1
outer_marsh_low	0	0	0 ↔ 0	1 ↔ 1	0 ↔ 0	0 ↔ 0	1 ↔ 1
outer_marsh_high	0	0	0 ↔ 0	**1 ↔ 1**	**0 ↔ 0**	**0 ↔ 0**	**1 ↔ 1**
outer_floodplain	0	0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
outer_cliff	1	1	1 ↔ 1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
inner_rockplatform	1	1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
inner_flood_dominance	0	0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
inner_ebb_dominance	1	1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
inner_subtidal_deposits	0	1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
inner_estuarywaves	1	0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
inner_sandflat	0	1	0 ↔ 0	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
inner_mudflat	0	1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
inner_marsh_low	0	1	**0 ↔ 0**	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
inner_marsh_high	0	0	0 ↔ 0	1 ↔ 1	1 ↔ 1	1 ↔ 1	1 ↔ 1
inner_floodplain	0	0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
inner_cliff	1	0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
bayhead_delta	1	1	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0	0 ↔ 0
river_discharge	1	1	1 ↔ 1	1 ↔ 1	1 ↔ 1	0 ↔ 0	0 ↔ 0
river_sand	1	1	0 ↔ 0	1 ↔ 1	1 ↔ 1	0 ↔ 0	0 ↔ 0
river_mud	0	1	1 ↔ 1	1 ↔ 1	1 ↔ 1	0 ↔ 0	0 ↔ 0

NB. Oscillating cyclical end points are indicated, with variables participating in cycle shown in **bold.**

Type 4: Spit-enclosed estuary. This system is forced with the imposition of a longshore wave power component and sediment supply that will lead to the formation of a protective spit. A river inflow is also imposed and distinguishes this type from a tidal inlet (type 7). The end point is cyclic between steps 8 and 11. Cyclic behaviour is confined to the outer estuary and involves interaction between high saltmarsh, prism and estuary waves. More generally, the evolution of this estuary morphology involves long-term infilling of both outer and inner estuary units, aided by the protection afforded by a spit. Ebb and flood deltas form within the estuary inlet and dunes are formed in association within the beaches and spit.

Type 5: Funnel-shaped drowned river valley. This is defined as a macro-tidal system and shows a tendency to infill, with the outer and inner estuary intertidal areas being sand and mud-dominated respectively. Although the total accommodation space is reduced over time, the tidal prism remains large since the outer estuary is wave-dominated and does not infill with saltmarsh. Linear banks replace the ebb and flood deltas of meso-tidal sediment-rich systems. The inner estuary saltmarsh leads to an ebb-dominated regime, although the outer estuary remains flood-dominant. This configuration evolves towards a cyclical state after steps 6 to 9. This cyclicity is due to interaction between waves, high saltmarsh and tidal asymmetry in the outer estuary.

Type 6: Embayment. As presently defined, embayments differ from the preceding funnel-shaped estuary type only in the lack of any single major river inflow. Given that the latter is not always significant in funnel-shaped systems, little difference in behaviour is to be expected. This is borne out by the model results, which show evolution towards a state characterised by outer linear banks and sand flats, and inner mudflats and saltmarsh. The evolution is cyclical after steps 6 to 9 due to interaction between waves, high saltmarsh and tidal asymmetry in the outer estuary. Embayments are modelled here as a macrotidal system. Under mesotidal or microtidal conditions, linear banks are replaced by tidal deltas, which is not necessarily realistic for an unconfined inlet geometry.

Type 7: Tidal inlet. Distinctive elements of a tidal inlet include the absence of any river flow and the tendency for one or more spits and tidal deltas at the estuary mouth. From the minimal set of initial conditions imposed in Table 10 such a system does indeed evolve. The end point is essentially the same as that for a spit-enclosed estuary and is cyclic between steps 8 and 11. Under micro-tidal conditions, however, there is provision in the Boolean functions for the inlet to block in the presence of littoral drift and the absence of significant river flow.

Overall, this analysis shows that the set of functions in Tables 7-9 are fairly effective in discriminating between the 7 generic estuary types, and that a range of evolutionary outcomes and behaviours emerge from relatively small variations in model input. The complexity of the morphodynamic behaviour varies between steady state and cyclical systems, and according to the extent and variety of sedimentary infilling.

If persistently inactive components are neglected, the evolutionary paths are easier to understand. The sequence of states for a fjord (Table 12) involves only 21 components and achieves steady state equilibrium in just 3 steps. The evolution of a fjard is a little more complex, involving 24 components over 5 steps (Table 13).

Table 12. Evolution of generic fjord towards steady state equilibrium

1	2	3	Step
1	1	1	ocean_waves
1	1	1	mesotidal
1	1	1	inlet
1	1	1	rock_basin
1	1	1	prism
1	1	1	accomm_space
1	1	1	bedrock
1	1	1	rock_sill
0	1	1	outer_rock_platform
0	1	1	outer_flood_dominance
0	1	0	outer_ebb_dominance
0	1	1	outer_estuary_swell
0	1	1	outer_estuarywaves
0	0	1	outer_cliff
0	1	1	inner_rockplatform
0	1	1	inner_ebb_dominance
0	0	1	inner_estuarywaves
0	1	1	inner_cliff
0	1	1	bayhead_delta
1	1	1	river_discharge
1	1	1	river_sand

NB. Step 1 corresponds to the initial conditions given in Table 11

Table 13. Evolution of generic fjard towards steady state equilibrium

1	2	3	4	5	Step
1	1	1	1	1	ocean_waves
1	1	1	1	1	mesotidal
1	1	1	1	1	inlet
1	1	1	1	1	rock_basin
1	1	1	0	0	prism
1	1	1	1	1	accomm_space
1	1	1	1	1	bedrock
0	1	1	1	1	outer_rock_platform
0	1	1	1	1	outer_flood_dominance
0	1	0	0	0	outer_ebb_dominance
0	1	1	1	1	outer_estuary_swell
0	1	1	1	0	outer_estuarywaves
0	0	1	1	1	outer_cliff
0	1	0	0	0	inner_rockplatform
0	1	1	1	1	inner_ebb_dominance
0	1	1	1	1	inner_subtidal_deposits
0	0	1	1	1	inner_sandflat
0	1	1	1	1	inner_mudflat
0	0	1	1	1	inner_marsh_low
0	1	1	0	0	inner_cliff
0	1	1	1	1	bayhead_delta
1	1	1	1	1	river_discharge
1	1	1	1	1	river_sand
1	1	1	1	1	river_mud

The evolution of a spit-enclosed estuary is more complex, involving 34 variables and the emergence of a cyclical end point after 11 steps (Table 14). The cycle length is 4 steps i.e. the interval between steps 8 and 11. Here, some artefacts of the synchronous updating are evident. At step 2, for example, the inner estuary appears to be both flood- and ebb-dominated. This is resolved over subsequent steps (including an intermediate state of neither flood- nor ebb-dominance) and ebb-dominance is finally established by step 6. Within the cycle, there is an interaction between prism, waves and high saltmarsh that owes more to the coding of the Boolean functions than a real cycle within the corresponding natural system. It is not immediately clear whether the cyclical behaviour evident in this and other estuary types is an emergent property of the system structure or whether it is a consequence of the synchronous updating scheme. This issue is considered further below.

Table 14. Evolution of generic spit-enclosed estuary towards cyclical equilibrium

1	2	3	4	5	6	7	**8**	**9**	**10**	**11**	Step
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	ocean_waves
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	longshore_power
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	littoral_sand
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	marine_sand
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	marine_mud
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	mesotidal
0	1	1	1	1	1	1	**1**	**1**	**1**	**1**	updrift_beach
0	0	1	1	1	1	1	**1**	**1**	**1**	**1**	spit
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	inlet
0	0	0	1	1	1	1	**1**	**1**	**1**	**1**	downdrift_beach
0	0	1	1	1	1	1	**1**	**1**	**1**	**1**	dunes
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	wind
1	1	0	1	1	0	1	**0**	**0**	**1**	**0**	prism
1	1	1	0	0	0	0	**0**	**0**	**0**	**0**	accomm_space
0	1	1	1	1	1	1	**1**	**1**	**1**	**1**	ebb_delta
0	1	1	1	1	1	1	**1**	**1**	**1**	**1**	flood_delta
0	1	1	1	1	1	0	**0**	**0**	**0**	**0**	outer_flood_dominance
0	1	0	0	0	0	0	**1**	**1**	**1**	**1**	outer_ebb_dominance
0	0	1	1	1	1	1	**1**	**1**	**1**	**1**	outer_subtidal_sands
0	1	1	0	0	0	0	**0**	**0**	**0**	**0**	outer_estuary_swell
0	1	1	0	1	1	0	**1**	**0**	**0**	**1**	outer_estuarywaves
0	0	1	1	0	1	1	**0**	**1**	**0**	**0**	outer_sandflat
0	1	0	0	1	1	1	**1**	**1**	**1**	**1**	outer_mudflat
0	0	0	0	0	1	1	**1**	**1**	**1**	**1**	outer_marsh_low
0	1	0	0	1	0	1	**1**	**0**	**1**	**1**	outer_marsh_high
0	1	1	1	1	1	1	**1**	**1**	**1**	**1**	inner_ebb_dominance
0	1	1	1	1	1	1	**1**	**1**	**1**	**1**	inner_subtidal_deposits
0	0	1	1	1	1	1	**1**	**1**	**1**	**1**	inner_sandflat
0	1	1	1	1	1	1	**1**	**1**	**1**	**1**	inner_mudflat
0	0	0	1	1	1	1	**1**	**1**	**1**	**1**	inner_marsh_low
0	0	1	1	1	1	1	**1**	**1**	**1**	**1**	inner_marsh_high
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	river_discharge
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	river_sand
1	1	1	1	1	1	1	**1**	**1**	**1**	**1**	river_mud

NB. Cyclical end sequence incorporating steps 8 to 11 indicated in **bold**.

6.4 Form-Process Feedback: Addition of Decay Term

A problem with the model as presently formulated is that transitions between states can result in unrealistic intermediate configurations. This can be attributed to the use of a discrete time step and to the synchronous updating of all state variables at each step. However in reality geomorphological system behaviour is conditioned by the temporal lag effects in respect of the morphological components. Some landforms, such as beaches, tend to respond rapidly to a change in forcing processes (perhaps over months or years), whilst others, such as saltmarshes, respond over longer timescales (perhaps decades or centuries). These lags tend to characterise the effect of a change in process on morphology. In contrast, changes in morphology (such as the breakdown of a spit) can often effect a near-instantaneous feedback on processes.

This aspect of system behaviour can be approximated with recourse to the decay term in the rate equation derived in Section 6.1 for the response of a state variable to a set of external parameters (equation 1). Any state variable can thus be assigned a decay term (or lag value) that controls the rate at which it changes state in response to inputs from connected state variables. If we retain synchronous updating at discrete time steps, then an obvious way to implement this is to assign arbitrary lag values (measured in discrete time steps) to selected components and to allow these components to change state only when subjected to persistent external stimulation for the duration of the lag. A spit, for example, might be assigned a lag of 3 steps. If the spit is subjected to external forcing that would result in its destruction, this must persist for at least 3 consecutive steps before its state variable will change state (in this case from logical 1 to 0). Multiple short-lived stimuli, in contrast, do not effect any change.

To evaluate the effect of such an implementation, decay terms can be applied to all of the morphological components that are free to adjust as the system evolves. Thus imposed features, rock basin, rock sill, fluvially-incised channel and cliffs were excluded from this assignment. Rapidly responding features were assigned an arbitrary decay of 3 steps, and slowly responding features a decay of 5 steps (Table 15).

State variable matrices for the evolution of each of the generic estuary types in the absence of engineering intervention or accelerated sea-level rise are presented in Tables 16-22.

Table 15. Assignment of decay terms to morphological variables

Coastal-Estuary Sub-System	**Decay**	
Coastal cliffs		none (imposed feature)
Barrier beach - up drift	rapid	
Spit	slow	
Inlet	rapid	
Barrier beach - down drift	rapid	
Coastal dunes	slow	
Outer Estuary Sub-System		
Ebb delta	rapid	
Flood delta	rapid	
Linear banks	rapid	
Rock sill		none (imposed feature)
Rock basin		none (imposed feature)
Fluvially-incised meandering channel		none (imposed feature)
Outer estuary rock platform (exposed)	rapid *	
Outer estuary subtidal deposits (incl channel sands)	rapid	
Outer estuary sand flat	rapid	
Outer estuary mud flat	rapid	
Outer estuary marsh - low	slow	
Outer estuary marsh - high	slow	
Outer estuary floodplain (0 if reclaimed)	rapid	
Outer estuary cliff		none (imposed feature)
Inner Estuary Sub-System		
Inner estuary rock platform (exposed)	rapid *	
Inner estuary subtidal deposits (channel sands/muds)	rapid	
Inner estuary sand flat	rapid	
Inner estuary mud flat	rapid	
Inner estuary marsh - low	slow	
Inner estuary marsh - high	slow	
Inner estuary floodplain (0 if reclaimed)	rapid	
Inner estuary cliff		none (imposed feature)
Bay head delta	slow	

* Rock platforms can be rapidly exposed by erosion of sediment cover but slower erosional development is not modelled.

Table 16. Evolution of generic fjord towards steady state

1	2	3	4	5	6	Step
1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	mesotidal
1	1	1	1	1	1	inlet
1	1	1	1	1	1	rock_basin
1	1	1	1	1	1	prism
1	1	1	1	1	1	accomm_space
1	1	1	1	1	1	bedrock
1	1	1	1	1	1	rock_sill
0	1	1	1	1	1	outer_rock_platform
0	1	1	1	1	1	outer_flood_dominance
0	1	1	1	1	1	outer_estuary_swell
0	1	1	1	1	1	outer_estuarywaves
0	0	1	1	1	1	outer_cliff
0	1	1	1	1	1	inner_rockplatform
0	0	1	1	1	1	inner_ebb_dominance
0	0	1	1	1	1	inner_estuarywaves
0	1	1	1	1	1	inner_cliff
0	0	0	0	0	1	bayhead_delta
1	1	1	1	1	1	river_discharge
1	1	1	1	1	1	river_sand

Table 17 Evolution of generic fjard towards steady state

1	2	3	4	5	6	7	8	9	10	Step
1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	mesotidal
1	1	1	1	1	1	1	1	1	1	inlet
1	1	1	1	1	1	1	1	1	1	rock_basin
1	1	1	1	1	1	1	1	0	0	prism
1	1	1	1	1	1	1	1	1	1	accomm_space
1	1	1	1	1	1	1	1	1	1	bedrock
0	1	1	1	1	1	1	1	1	1	outer_rock_platform
0	1	1	1	1	1	1	1	1	1	outer_flood_dominance
0	1	1	1	1	1	1	1	1	1	outer_estuary_swell
0	1	1	1	1	1	1	1	1	0	outer_estuarywaves
0	0	1	1	1	1	1	1	1	1	outer_cliff
0	1	1	1	0	0	0	0	0	0	inner_rockplatform
0	0	1	1	1	1	1	1	1	1	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	1	inner_subtidal_deposits
0	0	1	1	0	0	0	0	0	0	inner_estuarywaves
0	0	0	0	0	0	1	1	1	1	inner_sandflat
0	0	0	1	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	1	1	1	inner_marsh_low
0	1	1	1	1	1	1	1	0	0	inner_cliff
0	0	0	0	0	1	1	1	1	1	bayhead_delta
1	1	1	1	1	1	1	1	1	1	river_discharge
1	1	1	1	1	1	1	1	1	1	river_sand
1	1	1	1	1	1	1	1	1	1	river_mud

Table 18. Evolution of generic ria towards steady state

1	2	3	4	5	6	7	8	9	Step
1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	longshore_power
1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	mesotidal
0	0	0	1	1	1	1	1	1	updrift_beach
0	0	0	0	0	0	1	1	1	spit
1	1	1	1	1	1	1	1	1	inlet
1	1	1	1	1	1	1	1	1	fluvial_channel
0	0	0	0	0	0	0	0	1	dunes
1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	prism
1	1	1	1	1	1	1	1	1	accomm_space
1	1	1	1	1	1	1	1	1	bedrock
0	1	1	1	1	0	0	0	0	outer_rock_platform
0	1	1	1	1	1	1	1	1	outer_flood_dominance
0	1	1	1	1	1	1	1	1	outer_estuarywaves
0	0	0	0	1	1	1	1	1	outer_sandflat
0	0	1	1	1	1	1	1	1	outer_cliff
0	1	1	1	0	0	0	0	0	inner_rockplatform
0	0	1	1	1	1	1	1	1	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	inner_subtidal_deposits
0	0	1	1	0	0	0	0	0	inner_estuarywaves
0	0	0	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	1	1	inner_marsh_low
0	1	1	1	1	1	1	1	0	inner_cliff
1	1	1	1	1	1	1	1	1	river_discharge
1	1	1	1	1	1	1	1	1	river_mud

Table 19. Evolution of generic spit-enclosed estuary towards steady state

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Step
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	longshore_power
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	mesotidal
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	updrift_beach
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	spit
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inlet
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	downdrift_beach
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	prism
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	accomm_space
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	ebb_delta
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	flood_delta
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	outer_flood_dominance
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	outer_ebb_dominance
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	outer_subtidal_sands
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	outer_estuary_swell
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	outer_estuarywaves
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	outer_sandflat
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	outer_mudflat
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	outer_marsh_low
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	outer_marsh_high
0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	outer_floodplain
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_subtidal_deposits
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	inner_estuarywaves
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	inner_sandflat
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	inner_marsh_low
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	inner_marsh_high
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	river_discharge
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	river_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	river_mud

Table 20. Evolution of generic funnel-shaped estuary to steady state

1	2	3	4	5	6	7	8	9	10	11	12	13	Step
1	1	1	1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	macrotidal
0	0	0	1	1	1	1	1	1	1	1	1	1	updrift_beach
1	1	1	1	1	1	1	1	1	1	1	1	1	inlet
0	0	0	0	0	0	0	0	1	1	1	1	1	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	1	1	1	1	prism
1	1	1	1	1	1	1	0	0	0	0	0	0	accomm_space
0	0	0	1	1	1	1	1	1	1	1	1	1	linear_banks
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_flood_dominance
0	0	0	0	0	0	1	1	1	1	1	1	1	outer_subtidal_sands
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_estuary_swell
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_estuarywaves
0	0	0	1	1	1	1	1	1	1	1	1	1	outer_sandflat
0	1	1	1	1	1	1	1	0	0	0	0	0	inner_flood_dominance
0	0	0	0	0	0	0	0	0	0	1	1	1	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	1	1	1	1	inner_subtidal_deposits
0	0	1	1	0	0	0	0	0	0	0	0	0	inner_estuarywaves
0	0	0	0	0	0	1	1	1	1	1	1	1	inner_sandflat
0	0	0	1	1	1	1	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	0	0	0	0	0	1	inner_marsh_low
0	0	0	0	0	0	0	0	0	1	1	1	1	inner_marsh_high
1	1	1	1	1	1	1	1	1	1	1	1	1	river_discharge
1	1	1	1	1	1	1	1	1	1	1	1	1	river_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	river_mud

Table 21. Evolution of generic embayment towards steady state

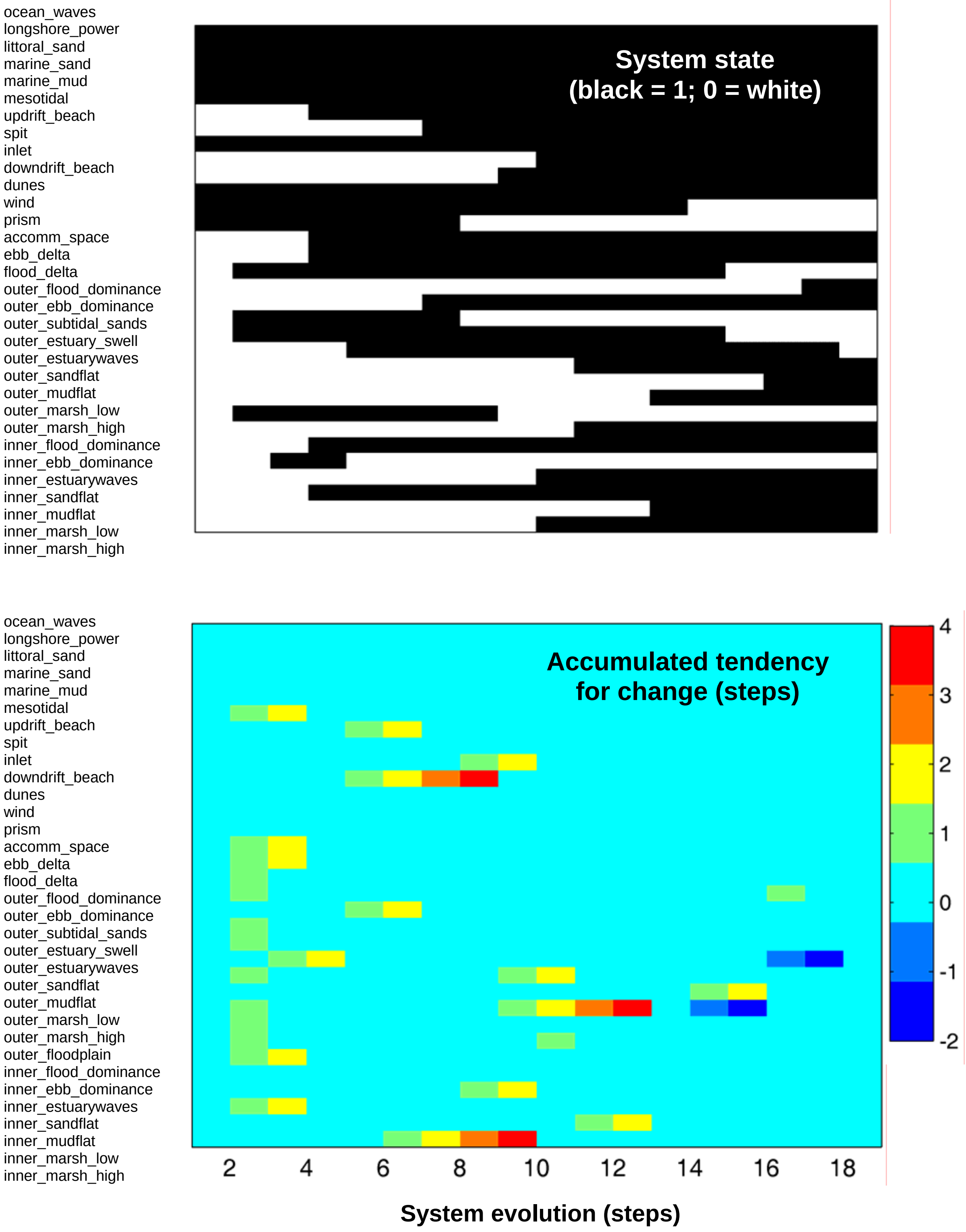
1	2	3	4	5	6	7	8	9	10	11	12	13	Step
1	1	1	1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	macrotidal
0	0	0	1	1	1	1	1	1	1	1	1	1	updrift_beach
1	1	1	1	1	1	1	1	1	1	1	1	1	inlet
0	0	0	0	0	0	0	0	1	1	1	1	1	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	1	1	1	1	prism
1	1	1	1	1	1	1	0	0	0	0	0	0	accomm_space
0	0	0	1	1	1	1	1	1	1	1	1	1	linear_banks
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_flood_dominance
0	0	0	0	0	0	1	1	1	1	1	1	1	outer_subtidal_sands
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_estuary_swell
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_estuarywaves
0	0	0	1	1	1	1	1	1	1	1	1	1	outer_sandflat
0	1	1	1	1	1	1	1	0	0	0	0	0	inner_flood_dominance
0	0	0	0	0	0	0	0	0	0	1	1	1	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	1	1	1	1	inner_subtidal_deposits
0	0	1	1	0	0	0	0	0	0	0	0	0	inner_estuarywaves
0	0	0	0	0	0	0	0	0	1	1	1	1	inner_sandflat
0	0	0	1	1	1	1	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	0	0	0	0	0	1	inner_marsh_low
0	0	0	0	0	0	0	0	0	1	1	1	1	inner_marsh_high

Table 22. Evolution of generic tidal inlet towards steady state

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Step
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	longshore_power
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	mesotidal
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	updrift_beach
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	spit
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inlet
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	downdrift_beach
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	prism
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	accomm_space
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	ebb_delta
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	flood_delta
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	outer_flood_dominance
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	outer_ebb_dominance
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	outer_subtidal_sands
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	outer_estuary_swell
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	outer_estuarywaves
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	outer_sandflat
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	outer_mudflat
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	outer_marsh_low
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	outer_marsh_high
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	inner_flood_dominance
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	inner_ebb_dominance
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	inner_estuarywaves
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_sandflat
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	inner_marsh_low
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	inner_marsh_high
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	inner_floodplain

The most obvious result of the added decay terms is that all of these generic estuary types now evolve towards a steady state endpoint. In the case of the fjord and fjard, the evolutionary end points are identical with or without decay terms, although the inclusion of the latter mean that system evolution now extends over more steps.

Some differences are apparent in the evolution of alluvial estuary types. In all cases, spurious oscillations are removed through the inclusion of decay terms and a more orderly sequence of landform development is generated. The effect of the decay terms can be seen more clearly by considering the evolution of the system 'memory' in parallel with the actual changes in state. Figure 15 shows this graphically for the case of a tidal inlet. The upper diagram shows the binary changes in system state. The lower diagram shows the accumulated tendencies for change, with positive and negative values corresponding to tendencies towards landform formation or destruction respectively. In one sense, this can be interpreted as the memory of the morphological components in response to changes resulting from the combined influences of other components and processes.



Upper diagram shows state matrix. Lower diagram shows accumulated tendency for variables to change (number of successive steps at which a variable is subject to positive or negative influence).

Figure 15. Graphical summary of system evolution for generic tidal inlet towards a steady state

6.5 External Forcing: Engineering Interventions and Accelerated Sea-Level Rise

The simulations presented so far relate to generic estuary types free to evolve towards a hypothetical natural equilibrium. However, provision is made in the Boolean functions for an estuary to be constrained in its morphological evolution by various human interventions. These include interferences with sediment supply (chiefly within the littoral drift system) and the imposition of fixed defences (with implications for the extent of saltmarsh and tidal prism). When present, interventions can result in a metastable equilibrium (e.g. Thorn and Welford, 1994) in which the system is stable for the duration of the constraint imposed by the intervention. If this constraint is removed the system evolves towards a different equilibrium condition.

Evolution of estuary morphology will also be influenced by changes in sea-level. Indeed all modern estuaries have formed as a result of marine inundation, with varying degrees of sedimentary infilling, since the last glaciation (Dalrymple *et al.*, 1992; Chappell and Woodroffe, 1995). In the provisional formulation of the model it is assumed that, prior to the significant human interventions of the last 200 years to so, the morphological evolution of UK estuaries incorporates a reduced influence of sea-level rise relative to that during the height of the mid-Holocene transgression. Accelerated sea-level rise associated with climate change will constitute a significant change in external forcing, however, and some of the indicative effects of this are incorporated in the model. These chiefly involve the tendency for intertidal landforms to diminish in extent where transgressive migration of estuary shorelines is checked by steep terrain or fixed defences ('coastal squeeze'). Sea-level rise also threatens the viability of defensive structures, such that these will fail unless they are actively maintained. The impact of accelerated sea-level rise on estuaries presently constrained in a metastable state is thus contingent upon management strategy.

A few of the possible intervention and sea-level scenarios are considered here, with reference to the behaviour of a tidal inlet. Table 23 shows the effect of imposing flood defences and then coastal protection and groynes, to a tidal inlet that has previously evolved to a natural equilibrium configuration. The result is a straightforward cessation of coastal cliff erosion and loss of the inner and outer saltmarsh, which is replaced by a protected floodplain. The loss of saltmarsh is delayed owing to the decay term associated with this landform. In the case of sea defence construction, the decay is unrealistic since the loss should be effectively instantaneous after imposition of a structure. There is no simple way to correct this in the current model, since no mathematical distinction is made between process-forcing and management interventions.

Table 24 shows the effect of subsequently imposing accelerated sea-level rise on the engineered inlet. In this scenario it is assumed that a 'hold the line' strategy is implemented, such that all structures are maintained and upgraded to cope with rising sea-level. The effect is the loss of various morphological components due to a sediment deficit in the littoral drift system and 'coastal squeeze' where transgressive migration of the intertidal is impeded by defences. If a switch is then made to a 'do nothing' strategy, the defences eventually reach the end of the life (defined by their decay term) at which time the estuary is free to evolve towards an unconstrained state (Table 25). In this particular case, this corresponds to a cyclical equilibrium involving the lower marsh units in both outer and inner estuary.

This set of scenarios shows that generalised management scenarios can be accommodated within the Boolean framework, and that broadly realistic estuary behaviour can be simulated using quite a simple generic rule base.

Table 23. Effect of imposing coastal protection, groynes, beach seawall, and estuary flood defences on a natural tidal inlet

1	2	3	4	5	6	7	Step
1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	longshore_power
1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	mesotidal
1	1	1	1	1	1	1	coastal_cliff
1	1	1	1	1	1	1	cliff_sand
1	1	1	1	1	1	1	cliff_mud
1	0	0	0	0	0	0	coastal_cliff_erosion
1	1	1	1	1	1	1	updrift_beach
1	1	1	1	1	1	1	spit
1	1	1	1	1	1	1	inlet
1	1	1	1	1	1	1	downdrift_beach
1	1	1	1	1	1	1	dunes
1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	groynes
1	1	1	1	1	1	1	coastal_protection
1	1	1	1	1	1	1	seawall
1	1	1	1	1	1	1	ebb_delta
1	1	1	1	1	1	1	flood_delta
1	1	1	1	1	1	1	outer_ebb_dominance
1	1	1	1	1	1	1	outer_subtidal_sands
1	1	1	1	1	1	1	outer_mudflat
1	1	1	1	1	1	1	outer_marsh_low
1	1	1	1	1	0	0	outer_marsh_high
1	1	1	1	1	1	1	outer_flood_defence
0	0	0	0	0	0	1	outer_floodplain
1	1	1	1	1	1	1	inner_ebb_dominance
1	1	1	1	1	1	1	inner_subtidal_deposits
1	1	1	1	1	1	1	inner_sandflat
1	1	1	1	1	1	1	inner_mudflat
1	1	1	1	1	1	1	inner_marsh_low
1	1	1	1	1	0	0	inner_marsh_high
1	1	1	1	1	1	1	inner_flood_defence
0	0	0	0	0	0	1	inner_floodplain

Table 24. Effect of accelerated SLR on heavily engineered natural tidal inlet of Table 23, assuming that a 'hold the line' strategy is pursued

1	2	3	4	5	6	7	8	9	10	11	12	Step
1	1	1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	longshore_power
1	1	1	1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	mesotidal
1	1	1	1	1	1	1	1	1	1	1	1	slr
1	1	1	1	1	1	1	1	1	1	1	1	coastal_cliff
1	1	1	1	1	1	1	1	1	1	1	1	cliff_sand
1	1	1	1	1	1	1	1	1	1	1	1	cliff_mud
1	1	1	0	0	0	0	0	0	0	0	0	updrift_beach
1	1	1	0	0	0	0	0	0	0	0	0	spit
1	1	1	1	1	1	1	1	1	1	1	1	inlet
1	1	1	1	1	1	0	0	0	0	0	0	downdrift_beach
1	1	1	1	1	1	1	1	1	1	1	0	dunes
1	1	1	1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	1	1	1	groynes
1	1	1	1	1	1	1	1	1	1	1	1	coastal_protection
1	1	1	1	1	1	1	1	1	1	1	1	seawall
1	1	1	1	1	1	1	1	1	1	1	1	hold_the_line
1	1	1	1	1	1	1	1	1	1	1	1	prism
1	1	1	1	1	1	1	1	1	1	1	1	ebb_delta
1	1	1	1	1	1	1	1	1	1	1	1	flood_delta
1	1	1	1	1	1	1	1	1	1	1	1	outer_flood_dominance
1	1	1	1	1	1	1	1	1	1	1	1	outer_subtidal_sands
0	0	0	0	1	1	1	1	1	1	1	1	outer_estuary_swell
1	1	1	1	1	1	1	1	1	1	1	1	outer_estuarywaves
1	1	1	1	1	1	1	1	1	1	1	1	outer_sandflat
1	1	1	1	1	1	1	0	0	0	0	0	outer_mudflat
1	1	1	0	0	0	0	0	0	0	0	0	outer_marsh_low
1	1	1	1	1	1	1	1	1	1	1	1	outer_flood_defence
1	1	1	1	1	1	1	1	1	1	1	1	inner_ebb_dominance
1	1	1	1	1	1	1	1	1	1	1	1	inner_subtidal_deposits
1	1	1	1	1	1	1	1	1	1	1	1	inner_sandflat
1	1	1	1	1	1	1	1	1	1	1	1	inner_mudflat
1	1	1	0	0	0	0	0	0	0	0	0	inner_marsh_low
1	1	1	1	1	1	1	1	1	1	1	1	inner_flood_defence

Table 25. Effect of abandoning the ‘hold the line’ strategy in favour of a ‘do nothing’ strategy

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	**23**	24	25	26	27	28	**29**	Step
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	longshore_power
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	mesotidal
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	slr
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	coastal_cliff
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	cliff_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	cliff_mud
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	coastal_cliff_erosion
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	updrift_beach
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	spit
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inlet
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	downdrift_beach
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	wind
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	groynes
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	coastal_protection
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	seawall
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	do_nothing
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	**0**	0	0	0	0	0	**0**	prism
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	ebb_delta
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	flood_delta
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	**0**	0	0	0	0	0	**0**	outer_flood_dominance
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	**1**	1	1	1	1	1	**1**	outer_ebb_dominance
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_subtidal_sands
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_estuary_swell
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	**0**	0	0	0	0	0	**0**	outer_estuarywaves
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**0**	0	0	0	0	0	**0**	outer_sandflat
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_mudflat
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	**1**	0	0	0	1	1	**1**	outer_marsh_low
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_marsh_high
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_flood_defence
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_floodplain
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_ebb_dominance
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_subtidal_deposits
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	**0**	0	0	0	0	0	**0**	inner_sandflat
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_mudflat
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	**0**	0	0	1	1	1	**0**	inner_marsh_low
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_marsh_high
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	inner_flood_defence
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	inner_floodplain

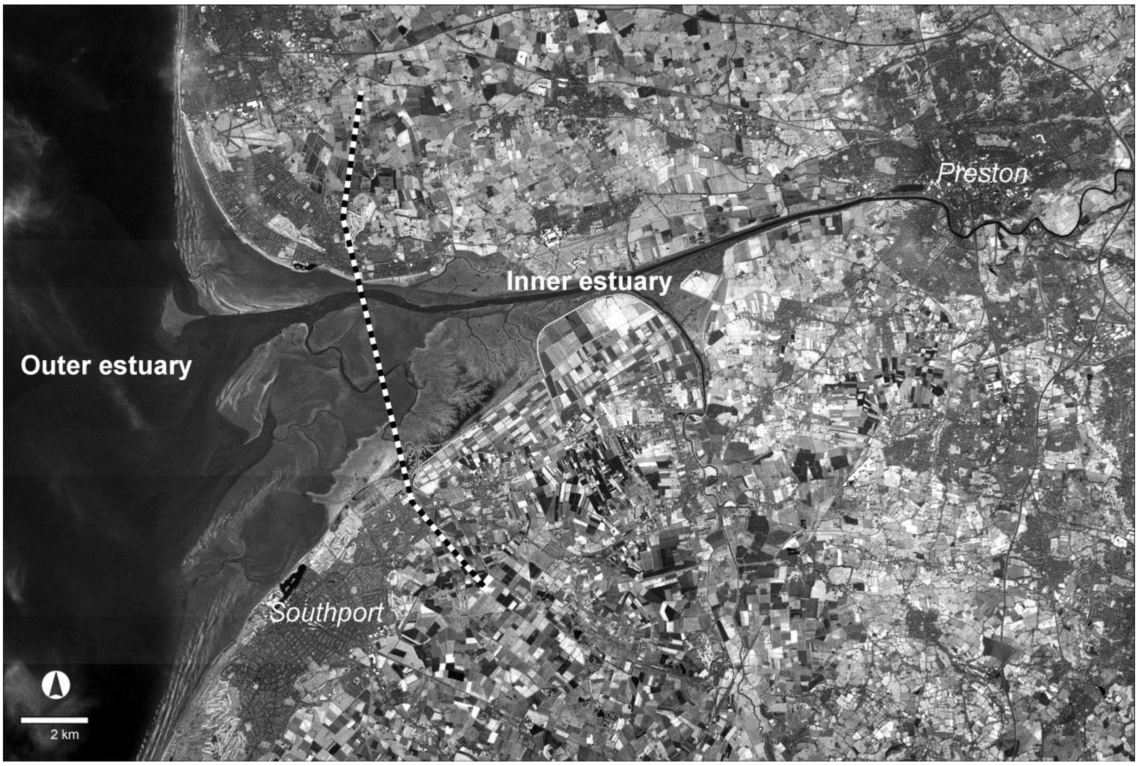
6.6 EstSim Boolean Model Applied to Historic Change: Ribble Estuary and Southampton Water

The model has been tested, without modification, on two case study estuaries. The Ribble (a funnel-shaped estuary) and Southampton Water (as an example of a spit-enclosed system) were selected as good examples of their respective types, and also due to their inclusion with earlier ERP Phase 1 studies (EMPHASYS, 2000).

6.6.1 The Ribble

The Ribble estuary, northwest England, is a classic example of a funnel-shaped coastal plain estuary (EstSim estuary type 5), although a significant portion of the system has been reclaimed over the last 200 years (Figure 16). With regard to the 'inner – outer' spatialisation, the contemporary Ribble estuary comprises:

- Outer estuary: extensive sand flat (upper and lower), beach, dune, trained channel;
- Inner estuary: extensive saltmarsh (upper and lower), narrow sand and mud flats, trained channel.



(Landsat ETM, 2002)

Figure 16. Delineation of outer and inner systems of the Ribble estuary

A series of detailed bathymetric charts span the last 150 years or so, and these have been used by van der Wal *et al.* (2002) to produce a synthesis of morphological change over this period. The estuary is subject to a macrotidal regime (mean spring tidal range of up to 8 m) and tidal flows are large compared to freshwater inflow from the River Ribble (and its main tributary the River Douglas). Wave energy along the coast and in the outer estuary is moderate. Although fluvial and estuarine sediment sources are fairly minimal, the potential offshore supply is extremely large: the system is thought to be infilling in the main from this marine source, with additional alongshore contributions from the open coasts to the north and south.

Table 26 encapsulates the major changes in the morphology of the estuary since the 1840s into a series of time periods (corresponding to the main bathymetric surveys) and approximate system states, defined using the enhanced set of Boolean variables proposed in Section 6.2. The Ribble has experienced a number of human-induced perturbations to system dynamics over this historical period, including reclamation, construction of training walls and intermittent dredging, and is hence a useful case study for this modelling application. Subjective judgements are required in order to abstract the subtleties of the documented changes into a form that is commensurate with the resolution of the model.

Table 26. Summary of historic changes in the Ribble estuary, northwest England

	pre-1840s Description	1904 Description	1951 Description	1994 Description	Remarks
External forcing					
Ocean waves	High	High	High	High	
Longshore wave power	Low	Low	Low	Low	
Marine sand supply	High	High	High	High	
Marine mud supply	High	High	High	High	
Tidal regime	Macrotidal	Macrotidal	Macrotidal	Macrotidal	
Wind regime for dunes	Favourable	Favourable	Favourable	Favourable	
Sea-level rise	Low	Low	Low	Accelerated	
River inflow	Yes + mud	Yes + mud	Yes + mud	Yes + mud	
Dredging	No	Yes	Yes	No	
System state variables					
Total tidal prism	High	Greatly reduced	Low	Low	19th century reclamation
Accommodation space	High	Reduced but dredged		Infilling, post-dredging	Dredged 1905-1980
Outer estuary waves	High	High	High	High	
Inner estuary waves	High	Low	Low	Low	
Outer estuary morphology					
Beach	Present	Present	Present	Present	
Dunes	Present	Present	Present	Present	
Spit	None	None	None	None	
Subtidal	Wide, deep	Narrow, shallow	Narrow, deep	Narrow, shallow	Dredged 1905-1980
Tidal delta	Linear banks	None	None	None	
Sand flat	small, low	extensive low	extensive high	extensive high	Extensive sandflat accretion
Mud flat	absent	absent	absent	absent	
Salt marsh	absent	absent	absent	small	Spartina planted 1930s, spreads 1960s-70s
Flood defences	None	Yes	Yes	Yes	19th century reclamation
Floodplain	None	Yes	Yes	Yes	
Inner estuary morphology					
Subtidal	Wide, shallow	Narrow, shallow	Narrow, deep	Narrow, shallow	Dredged 1905-1980
Sand flat	Extensive	Medium	Small	Absent	Sand flat replaced by mud flat, then marsh
Mud flat	Small	Small	Medium	Medium	Sud flat expands at expense of sand flat
Salt marsh	High marsh	None	Low marsh	Low marsh	Spartina planted 1930s, spreads 1960s-70s
Flood defences	None	Yes	Yes	Yes	19th century reclamation
Floodplain	None	Yes	Yes	Yes	

(Largely adapted from van der Wal et al. (2002)

In essence, Table 26 charts a transition from a rather wide system dominated by extensive low sand flats, to a narrower and deeper system which exhibits a clear transition from an intertidal sand-dominated outer estuary to mixed sediment, saltmarsh dominated inner estuary. Reclamation throughout the 19th century significantly reduced the tidal prism and intertidal area within the inner estuary: subsequent sedimentation, particularly during periods of dredging in the mid-20th century, preferentially infilled the reduced intertidal zone leading to an expanse of saltmarsh at the expense of intertidal flat. The channel in this region has been trained throughout the period considered here, and is now flanked by narrow strips of sand and mudflat. In the outer estuary, training walls constructed in the late 19th and early 20th century have replaced the multiple and dynamic character with a single, relatively stable channel, resulting in accretion across the intertidal flats. The cessation of dredging in the latter half of the 20th century has caused some local erosion of sand banks as the channel has started to accommodate some of the estuarine sediment volume.

Evolutionary end points predicted by the EstSim model represent long-term equilibrium configurations. An initial difficulty encountered during evaluation of model output against the documented change for the Ribble concerns the extent to which the real world system can be considered to have achieved equilibrium. Van der Wal *et al.* (2002) suggest that given the abundance of sediment and low rate of relative sea-level rise, the gross morphology of the Ribble was probably close to a state of dynamic equilibrium at the beginning of the 19th century. After that date, a succession of human interventions (reclamation, dredging, planting of saltmarsh, cessation of dredging) occurred at intervals that were probably much shorter than the relaxation time towards new equilibrium states. It is likely, therefore, that we need to scrutinise not just the simulated end states but also their evolutionary trajectories. This needs to be done with care, bearing in mind the observation made in the previous section that some intermediate states are likely to be artefacts of the synchronous updating of the Boolean network.

As a first step, pre-1840 process forcing was approximated by high onshore wave energy, abundant sediment and no human intervention and the model allowed to run to equilibrium (see Table 27). A steady state emerges after 13 steps and this corresponds closely to the natural state described by van der Wal *et al.* (2002) and summarised in Table 26. Overall, the picture during this epoch is one of sedimentary infilling, with sandy intertidal deposits dominating the outer estuary (where wave action remains significant) and mud flat and saltmarsh being more widespread in the inner estuary. The fluvial mud input is trapped within the inner estuary and was known to be a cause of siltation within Preston Docks (van der Wal *et al*., 2002). The general behaviour of this system in its pre-1840 state is thus captured very well.

The major human interventions during the 19th century were reclamation of a large part of the inner estuary intertidal and a smaller part of the outer estuary along the southern shore (with a reduction in the area of the tidal floodplain and saltmarsh). The result of this change imposed on the endpoint of the pre-1840 run is shown in Table 28. The main effect is a large reduction in prism, with loss of high saltmarsh. In the outer estuary, there is now a tendency for low marsh to form, although this is unstable and results in a cyclical equilibrium after 6 to 12 steps. A shift towards ebb dominance in the outer estuary is not very realistic (the Ribble seems to have remained flood dominated), and results from what is probably a significant over-estimation of the reduction in tidal prism through reclamation.

Table 27. Simulated evolution of Ribble Estuary for pre-1840 epoch

1	2	3	4	5	6	7	8	9	10	11	12	13	Step
1	1	1	1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	macrotidal
1	1	1	1	1	1	1	1	1	1	1	1	1	updrift_beach
1	1	1	1	1	1	1	1	1	1	1	1	1	inlet
0	0	0	0	0	1	1	1	1	1	1	1	1	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	1	1	1	1	prism
1	1	1	1	1	1	1	0	0	0	0	0	0	accomm_space
0	0	0	1	1	1	1	1	1	1	1	1	1	linear_banks
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_flood_dominance
0	0	0	0	0	0	1	1	1	1	1	1	1	outer_subtidal_sands
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_estuary_swell
0	1	1	1	1	1	1	1	1	1	1	1	1	outer_estuarywaves
0	0	0	1	1	1	1	1	1	1	1	1	1	outer_sandflat
0	1	1	1	1	1	1	1	0	0	0	0	0	inner_flood_dominance
0	0	0	0	0	0	0	0	0	0	1	1	1	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	1	1	1	1	inner_subtidal_deposits
0	0	1	1	0	0	0	0	0	0	0	0	0	inner_estuarywaves
0	0	0	0	0	0	0	0	0	1	1	1	1	inner_sandflat
0	0	0	1	1	1	1	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	0	0	0	0	0	1	inner_marsh_low
0	0	0	0	0	0	0	0	0	1	1	1	1	inner_marsh_high
1	1	1	1	1	1	1	1	1	1	1	1	1	river_discharge
1	1	1	1	1	1	1	1	1	1	1	1	1	river_mud

Table 28. Simulated evolution of Ribble Estuary for 1840 - 1904 epoch

1	2	3	4	5	**6**	7	8	9	10	11	**12**	Step
1	1	1	1	1	**1**	1	1	1	1	1	**1**	ocean_waves
1	1	1	1	1	**1**	1	1	1	1	1	**1**	littoral_sand
1	1	1	1	1	**1**	1	1	1	1	1	**1**	marine_sand
1	1	1	1	1	**1**	1	1	1	1	1	**1**	marine_mud
1	1	1	1	1	**1**	1	1	1	1	1	**1**	macrotidal
1	1	1	1	1	**1**	1	1	1	1	1	**1**	updrift_beach
1	1	1	1	1	**1**	1	1	1	1	1	**1**	inlet
1	1	1	1	1	**1**	1	1	1	1	1	**1**	dunes
1	1	1	1	1	**1**	1	1	1	1	1	**1**	wind
1	0	0	0	0	**0**	0	0	0	0	0	**0**	prism
1	1	1	1	1	**1**	1	1	1	1	1	**1**	linear_banks
1	1	0	0	0	**0**	0	0	0	0	0	**0**	outer_flood_dominance
0	0	0	0	1	**1**	1	1	1	1	1	**1**	outer_ebb_dominance
1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_subtidal_sands
1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_estuary_swell
1	1	0	0	0	**0**	0	0	0	0	0	**0**	outer_estuarywaves
1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_sandflat
0	0	0	0	0	**1**	1	1	1	1	1	**1**	outer_mudflat
0	0	0	0	0	**1**	1	1	0	0	0	**1**	outer_marsh_low
1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_flood_defence
0	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_floodplain
1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_ebb_dominance
1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_subtidal_deposits
1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_sandflat
1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_mudflat
1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_marsh_low
1	1	1	1	1	**0**	0	0	0	0	0	**0**	inner_marsh_high
1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_flood_defence
0	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_floodplain
1	1	1	1	1	**1**	1	1	1	1	1	**1**	river_discharge
1	1	1	1	1	**1**	1	1	1	1	1	**1**	river_mud

NB. Evolution becomes cyclical between steps 6 and 12.

Following the next major intervention, dredging and training of the estuary after 1904 (Table 29), the accommodation space is enlarged and the outer estuary becomes flood-dominant once more. The behaviour in this epoch is not entirely consistent within the documented inner estuary changes, which involved a rather slower transition back to a muddy intertidal and re-establishment of saltmarsh accelerated by planting after the 1950s.

Table 29. Simulated evolution of Ribble Estuary for 1904-1980 epoch

1	2	3	4	**5**	6	7	8	9	10	**11**	Step
1	1	1	1	**1**	1	1	1	1	1	**1**	ocean_waves
1	1	1	1	**1**	1	1	1	1	1	**1**	littoral_sand
1	1	1	1	**1**	1	1	1	1	1	**1**	marine_sand
1	1	1	1	**1**	1	1	1	1	1	**1**	marine_mud
1	1	1	1	**1**	1	1	1	1	1	**1**	macrotidal
1	1	1	1	**1**	1	1	1	1	1	**1**	updrift_beach
1	1	1	1	**1**	1	1	1	1	1	**1**	inlet
1	1	1	1	**1**	1	1	1	1	1	**1**	dunes
1	1	1	1	**1**	1	1	1	1	1	**1**	wind
0	1	1	1	**1**	1	1	1	1	1	**1**	accomm_space
1	1	1	1	**1**	1	1	1	1	1	**1**	linear_banks
0	0	1	1	**1**	1	1	1	1	1	**1**	outer_flood_dominance
1	1	1	1	**0**	0	0	0	0	0	**0**	outer_ebb_dominance
1	1	1	0	**0**	0	0	0	0	0	**0**	outer_subtidal_sands
1	1	1	1	**1**	1	1	1	1	1	**1**	outer_estuary_swell
1	1	1	1	**1**	1	1	1	1	1	**1**	outer_sandflat
1	1	1	1	**1**	1	1	1	1	1	**1**	outer_mudflat
0	0	0	1	**1**	1	0	0	0	1	**1**	outer_marsh_low
1	1	1	1	**1**	1	1	1	1	1	**1**	outer_flood_defence
1	1	1	1	**1**	1	1	1	1	1	**1**	outer_floodplain
1	1	1	1	**1**	1	1	1	1	1	**1**	outer_dredging
1	1	1	1	**1**	1	1	1	1	1	**1**	inner_ebb_dominance
1	1	1	0	**0**	0	0	0	0	0	**0**	inner_subtidal_deposits
1	1	1	1	**1**	1	1	1	1	1	**1**	inner_sandflat
1	1	1	1	**1**	1	1	1	1	1	**1**	inner_mudflat
1	1	1	1	**1**	1	1	1	1	1	**1**	inner_marsh_low
1	1	1	1	**1**	1	1	1	1	1	**1**	inner_flood_defence
1	1	1	1	**1**	1	1	1	1	1	**1**	inner_floodplain
1	1	1	1	**1**	1	1	1	1	1	**1**	inner_dredging
1	1	1	1	**1**	1	1	1	1	1	**1**	river_discharge
1	1	1	1	**1**	1	1	1	1	1	**1**	river_mud

Nb. Evolution becomes cyclical between steps 5 and 11.

The final epoch sees the cessation of dredging but also the imposition of accelerated sea-level rise (Table 30). In the absence of a ‘hold the line’ strategy, the flood defences eventually fail and the system reverts to a more natural state. This comprises elements of the generic funnel shaped estuary (wave-dominated outer estuary with sandflats; tide-dominated inner estuary with saltmarsh), with cyclicity arising from the narrowing of the space available for saltmarsh.

Table 30. Simulated evolution of Ribble Estuary for post-1980s with imposition of accelerated SLR

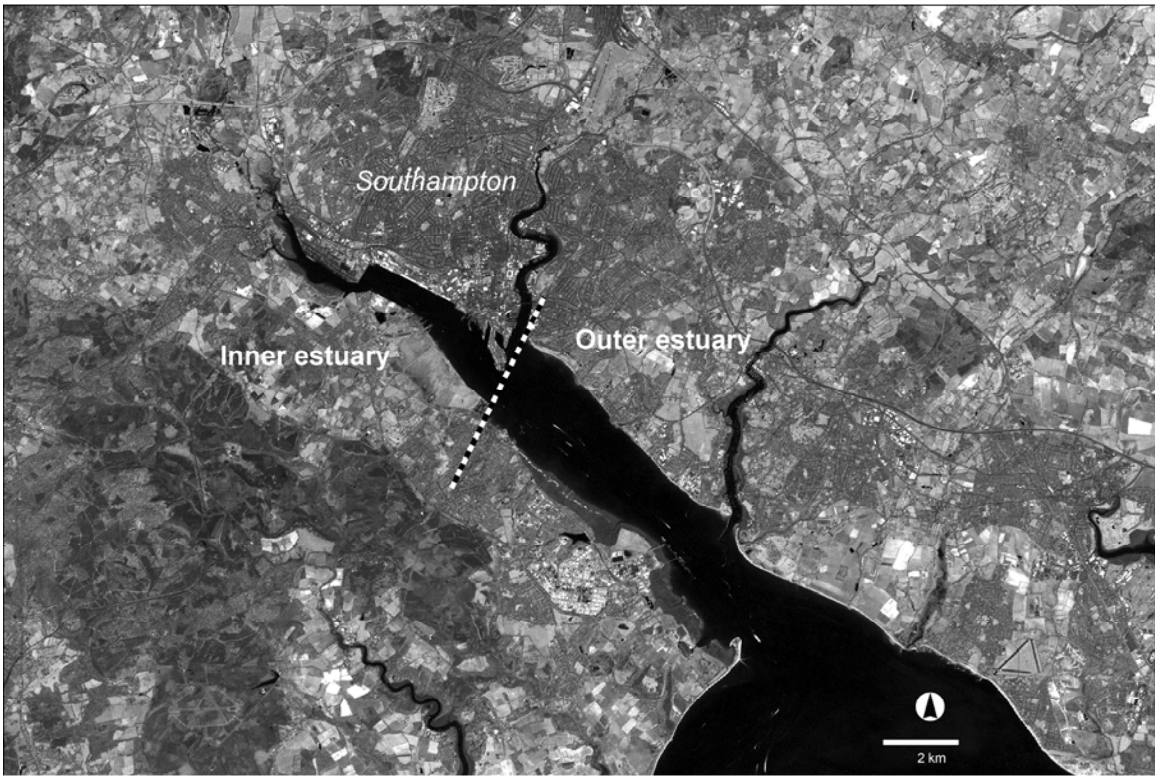
1	2	3	4	5	6	7	8	9	10	11	12	13	14	**15**	16	17	18	19	20	**21**	step
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	macrotidal
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	slr
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	updrift_beach
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inlet
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	wind
0	0	0	0	0	0	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	prism
1	0	0	0	0	0	0	1	1	1	1	0	0	0	**0**	0	0	0	0	0	**0**	accomm_space
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	linear_banks
1	1	0	0	0	0	0	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_flood_dominance
0	0	0	0	1	1	1	1	1	0	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_ebb_dominance
0	0	0	0	0	0	0	0	0	0	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_subtidal_sands
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_estuary_swell
0	0	0	0	0	0	0	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_estuarywaves
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	outer_sandflat
1	1	1	1	1	1	1	1	1	1	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_mudflat
1	1	1	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_marsh_low
1	1	1	1	1	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_flood_defence
1	1	1	1	1	1	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	outer_floodplain
0	0	0	0	0	0	0	0	1	1	1	1	0	0	**0**	0	0	0	0	0	**0**	inner_flood_dominance
1	1	1	1	1	1	1	1	1	1	0	0	0	0	**1**	1	1	1	1	1	**1**	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_subtidal_deposits
1	1	1	1	1	0	0	0	0	0	0	0	0	1	**1**	1	1	1	1	1	**1**	inner_sandflat
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_mudflat
1	1	1	0	0	0	0	0	0	0	0	0	0	1	**1**	1	0	0	0	1	**1**	inner_marsh_low
0	0	0	0	0	0	0	0	0	0	1	1	1	1	**1**	1	1	1	1	1	**1**	inner_marsh_high
1	1	1	1	1	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	inner_flood_defence
1	1	1	1	1	1	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	**0**	inner_floodplain
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	river_discharge
1	1	1	1	1	1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	**1**	river_mud

Nb. Evolution becomes cyclical between steps 15 and 21.

6.6.2 Southampton Water

Southampton Water, on the south coast of England, is an example of a spit-enclosed estuary (EstSim estuary type 4). A deep and relatively straight estuary channel receives inflows from three rivers, the Itchen and Test towards the head, and the Hamble close to the mouth on the eastern margin (Figure 17).

The system is mesotidal (spring tidal range approximately 4 m at Southampton), with complex tidal propagation from the Solent being associated with double high waters at spring tides. Spring tidal prism is approximately 1 x 10^8 m^3 (Long *et al.*, 2000). In terms of velocity asymmetry, the estuary is ebb dominated throughout, although this diminishes towards the estuary head.



(Landsat ETM, 2002)

Broken line indicates suggested division into inner and outer estuary sub-systems.

Figure 17. Aerial view of Southampton Water

A behavioural statement case study for Southampton Water is included in FD2117 Project Record PR2 '*EstSim Behavioural Statements Report* 2' (EstSim Consortium, 2004). This emphasises several important aspects of the estuary geomorphology:

- The role of Calshot Spit in narrowing the inlet and limiting wave action within the outer estuary.
- A positive sediment budget, with a tendency for the estuary to infill with fine sediment, largely of marine origin.

- Relatively minor contributions to overall estuary hydrodynamics and sediment budget from the fluvial inputs.
- The occurrence of mudflat and saltmarsh in both outer and inner estuary but with evidence of an erosional tendency and some topographic constraint upon the ability of intertidal environments to undergo a natural transgressive migration in response to sea-level rise (especially in the outer estuary).
- Historic reclamation of much of the outer and inner estuary intertidal.
- The importance of dredging in maintaining a deep outer estuary channel.

This case study is less suited to a sequential historical analysis of the kind undertaken for the Ribble. However, it presents a useful opportunity to evaluate the generic Boolean functions against a system that is extremely well documented in terms of its physical processes and which also exhibits some differences from the idealised spit-enclosed estuary considered previously. One problem is the presence of steep catchment topography that constrains the long-term adjustment of the outer estuary. This is approximated here by the imposition of bedrock (implying a resistant geology and elevated catchment topography) but without an inherited fluvially-incised channel. This allows the formation of cliffs, which will then lead to ‘coastal squeeze’ under an accelerated sea-level rise scenario. In this sense, the system lies someway between the ria and spit-enclosed generic types.

Table 31 shows the evolution of such a system towards a steady state. The littoral drift system leads to the formation of a spit, but the absence of tidal deltas (owing to dredging of the outer estuary) limits sediment bypassing across the inlet such that no downdrift beach forms. The wave action is reduced by the spit and muddy depositional environments form within the narrowed intertidal accommodation space of both inner and outer estuary. The whole system is ebb dominated, by virtue of the dredging and infilling of the limited intertidal. Both prism and accommodation space are shown as greatly reduced. In the case of the latter, the removal of tidal floodplain by reclamation outweighs the effect of continued outer estuary dredging in the present model formulation.

If accelerated sea-level rise is imposed (Table 32), then a characteristic ‘coastal squeeze’ emerges. In the assumed absence of a ‘hold the line’ strategy the reclaimed floodplain eventually gives way to tidal flat and a cyclical alternation between wave-dominated sandflat and a low saltmarsh. The estuary also oscillates between ebb and flood dominance under this scenario.

As with the Ribble case study, the analysis of Southampton Water shows that a substantially correct depiction of gross estuary properties can be obtained with an unaltered generic rule base. However, there are subtle estuary-specific aspects of inherited morphology, sediment transport, hydrodynamics, and intervention history that would require customisation of the model functions to elucidate more fully. Some features, such as the distinctive tidal hydrodynamics of the Solent – Southampton Water system, are almost certainly beyond the capability of a generalised model to resolve.

Table 31. Simulated evolution of Southampton Water with no accelerated SLR

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Step
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	ocean_waves
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	longshore_power
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	littoral_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	marine_sand
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	marine_mud
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	mesotidal
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	updrift_beach
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	spit
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inlet
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	dunes
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	wind
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	prism
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	accomm_space
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	bedrock
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	outer_rock_platform
0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	outer_flood_dominance
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	outer_ebb_dominance
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	outer_estuary_swell
0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	outer_estuarywaves
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	outer_sandflat
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	outer_mudflat
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	outer_marsh_low
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	outer_flood_defence
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	outer_floodplain
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	outer_cliff
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	outer_dredging
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	inner_rockplatform
0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	inner_flood_dominance
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	inner_ebb_dominance
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_subtidal_deposits
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	inner_estuarywaves
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_mudflat
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	inner_marsh_low
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_flood_defence
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	inner_floodplain.

NB. System reaches steady state after 19 steps.

Table 32. Continued evolution of Southampton Water with imposition of accelerated SLR

1	2	3	4	5	6	7	8	9	**10**	11	12	13	14	15	16	17	18	**19**	Step
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	ocean_waves
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	longshore_power
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	littoral_sand
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	marine_sand
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	marine_mud
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	mesotidal
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	slr
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	updrift_beach
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	spit
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	inlet
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	dunes
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	wind
0	0	0	0	1	1	1	1	1	**1**	1	0	0	0	1	1	1	1	**1**	prism
0	0	0	0	0	1	1	1	1	**1**	1	1	0	0	0	1	1	1	**1**	accomm_space
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	bedrock
0	0	0	0	0	1	1	1	1	**1**	1	1	1	0	0	1	1	1	**1**	outer_flood_dominance
1	1	1	1	1	1	1	0	0	**0**	0	0	0	0	0	1	1	0	**0**	outer_ebb_dominance
0	0	0	0	0	1	1	1	1	**1**	1	1	0	0	0	1	1	1	**1**	outer_estuarywaves
0	0	0	0	0	0	0	0	1	**1**	1	1	1	1	1	0	0	0	**1**	outer_sandflat
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	outer_mudflat
1	1	1	0	0	0	0	0	0	**0**	1	1	1	0	0	0	0	0	**0**	outer_marsh_low
1	1	1	1	1	0	0	0	0	**0**	0	0	0	0	0	0	0	0	**0**	outer_flood_defence
1	1	1	1	1	1	0	0	0	**0**	0	0	0	0	0	0	0	0	**0**	outer_floodplain
0	0	0	0	0	0	0	1	1	**1**	1	0	0	0	0	0	1	1	**1**	outer_cliff
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	outer_dredging
0	0	0	0	0	0	1	1	1	**1**	1	1	0	0	0	0	1	1	**1**	inner_flood_dominance
1	1	1	1	1	1	1	1	0	**0**	0	0	0	0	1	1	1	1	**0**	inner_ebb_dominance
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	inner_subtidal_deposits
1	1	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	1	**1**	inner_mudflat
1	1	1	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	**0**	inner_marsh_low
1	1	1	1	1	0	0	0	0	**0**	0	0	0	0	0	0	0	0	**0**	inner_flood_defence
1	1	1	1	1	1	0	0	0	**0**	0	0	0	0	0	0	0	0	**0**	inner_floodplain
0	0	0	0	0	0	0	1	1	**1**	1	1	1	1	1	1	1	1	**1**	inner_cliff

Nb. System becomes cyclical after steps 10 to 19.

6.7 Towards the Development of a GUI-Based Tool

The prototype simulator is implemented in *MATLAB* and presently supports:

- Representation of estuary system influence diagrams for the generic estuary types defined in EstSim Project Record PR2 using standardised sets of variables and Boolean functions.
- Initial condition-based simulation of system evolutionary trajectories.
- State variable space simulation, based on either analysis of all possible states (small *N* systems) or statistical sampling of these states (large *N* systems), that can be used to identify and classify equilibrium states, and derive various measures of system complexity.

The Boolean functions are evaluated directly as mathematical expressions by *MATLAB*, thereby avoiding the need for the user to program any of the model functionality using a high-level language. The development work presented in this report has been undertaken using a minimal set of system components and a universal set of Boolean functions to cover all estuary types. Whilst this is sufficient for proof of concept testing more detailed analysis of specific estuary types or the response of a case-study estuary to particular combinations of environmental forcing or management intervention would ideally involve further customisation of both the variable and function sets.

The simulator is written in MATLAB's scripting language and uses two kinds of external steering file. The first, the *estuary definition file*, specifies the set of morphological and process components and their initial states for a given estuary (or generic estuary type). A typical estuary definition file is provided in Appendix 4. The second, a *Boolean function library*, contains the Boolean functions that characterise the behaviour of all the possible components. A single function library can be used for all simulations, or custom libraries can be developed for particular applications.

A guiding principle is that all of the model logic (that is, the formalisation of geomorphological knowledge) resides in the external steering files (i.e. in the set of system variables and their Boolean functions). The simulator code merely loads the Boolean function library along with a user-selected estuary system definition file, and computes the evolutionary behaviour of the system based upon the supplied initial states. Provided that some basic formatting rules are followed in the creation of the steering files, the user can extend the variable set, modify the Boolean functions and create any number of scenarios without modifying the simulator code itself. A key point here is that the methodology is extensible and has the potential to be extended to more challenging modelling tasks.

A more interactive interface to the simulator has been developed using *MATLAB's* built-in GUI tools. This provides file selection dialog boxes, access to various run control parameters, the ability to switch between scenario and state-space analysis, and the display of model results. Figure 18 illustrates a layout for an early version of the GUI and its relation to underlying model code, the Boolean function library and a set of estuary definition files.

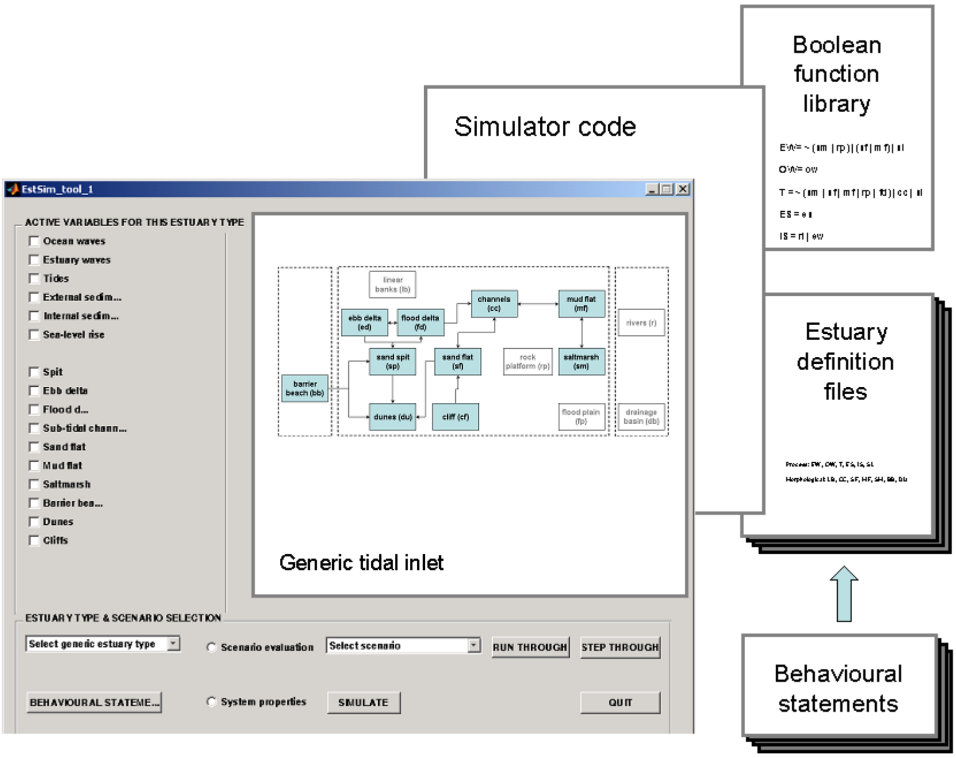


Figure 18: Illustration of early prototype GUI for standalone EstSim tool developed in *MATLAB*

The GUI-based tool will be compiled into an easy to use application that can be freely distributed to a wider range of users who do not have access to MATLAB software. This, together with the full MATLAB source code, will be made available via the internet (www.geog.ucl.ac.uk/ceru/estsim).

As a simple standalone tool, the MATLAB-based simulator is intended to supplement a web-based simulator (as envisaged in section 5 of this report) that is being developed under Objective 6 of FD2117 by Discovery Software (www.discoverysoftware.co.uk) and also to provide a basis for further experimentation by interested researchers.

6.8 Strengths and Weaknesses of Network Models Formulated Using Boolean Functions

A major advantage of network-based models is that they allow inferences concerning the behaviour of large and complex geomorphic systems that can be reasonably well-defined in terms of a set of components but which remain ill-defined in terms of the mechanistic interactions between these components. This is very much in accordance with the arguments advanced by Capobianco *et al.* (1999). However, the QR methodologies that they advocate depend on rigorous analysis of the feedbacks (or ‘causal loops’) in a system and this demands considerable expertise in quantitative as well as qualitative modelled (e.g. Wolstenholme, 1999). In contrast, a Boolean approach makes few assumptions about the underlying system behaviour and only requires the modeller to have sufficient experience to translate geomorphological understanding into a series of logical functions. Aspects of system behaviour are computed directly from the natural Boolean expressions using any numerical computation software or programming language that supports the small set of standard logical operators. The emphasis is placed very firmly on the formalisation of qualitative knowledge through system and influence diagrams, and the consensus (or otherwise) amongst expert geomorphologists regarding the translation of the latter into functional form. A Boolean network model thus provides a good basis for comparative evaluation of alternative conceptual models and for identifying aspects of system behaviour that are sensitive to differences in interpretation or scientific opinion.

Like all modelling methodologies, the Boolean network approach also has its limitations. Some of the consequences of adopting a discrete binary logic have been touched upon in section 6.1 of this report. In particular, the assumption that influences between system variables can be thresholded to yield meaningful binary states and that these influences act together with a similar weight and in synchrony is clearly restrictive and sometimes difficult to justify in terms of what is known of the underlying physical and biological processes. The introduction of decay terms for morphological components is shown here to yield more realistic evolutionary behaviour and it would be interesting to investigate differences in the properties of synchronous and asynchronous models in more detail. Alternatively, continuous mathematical functions and associated sets of influence weights (of the kind envisaged by Seymour, 2004) could be substituted for the discrete Boolean logic. This would allow representation of gradational changes but would require the a priori specification of influence weights for each of the component variables. Calibration of such a model might be difficult for a large system, though some form of inverse modelling might be used to infer sets of weights from historical sequences of system state.

Conversion of qualitative geomorphological understanding into Boolean function form can also be problematic. The functions chosen must be complex enough to accommodate a worthwhile range of system behaviour, yet they must also be consistent in their implementation in order that the predicted behaviour constitutes an emergent property of the system rather than a consequence of logical inconsistency in some of its defining functions. For interactions of a few components, recourse can be made to the logical truth table or binary decision tree and potentially undesirable behaviour identified at the level of the individual function (see, for example, Figures 13 and 14). Richer variable sets bring the potential for more complex functions, however, and the resulting truth tables may be quite large (recalling that there will be 2^N potential states where N is the number of components). Techniques such as ordering of the variables, deletion of redundant nodes and sharing of equivalent sub-graphs can help here. However, there is likely to be a trade-off between the complexity with which we define a system in terms of a finite set of components and the tractability of defining a logically-consistent set of constituent functions. Further experimentation is needed to find this optimum.

The above limitations notwithstanding, the results obtained with a simple proof of concept Boolean model appear sufficiently interesting to prompt further work in this vein. Future efforts should perhaps focus upon the following areas:

1. The evaluation of more refined variable sets and the development of approaches (and software tools) for the development and testing of complex yet logically rigorous Boolean functions.
2. Further experimentation with linked sub-systems as a means of minimising the complexity of individual functions whilst increasing the ability of a Boolean model to resolve the subtleties of estuary system behaviour.
3. Investigation of the operator variance associated with each stage of the modelling process (i.e. system mapping, influence diagram construction, formalisation of knowledge into model functions).
4. Experimentation with variable decay terms to encompass a broader variety of non-synchronous behaviour.
5. Experimentation with non-Boolean representations that could be accommodated within the GUI framework developed for EstSim (e.g. the use of continuous response functions and associated sets of influence weights).
6. Comparison of Boolean network model results with those obtained using a QR methodology using software suited to environmental systems applications. The recently released GARP3 package may have potential here (e.g. Tullos *et al*., 2004).

7. CONCLUSIONS AND RECOMMENDATIONS

A qualitative approach provides an alternative basis for modelling the behaviour of estuary and coastal systems, as abstracted at the scales of most relevance to the tackling of estuary management questions. Emergent properties of system behaviour may be revealed, including the existence of unexpected sensitivities to change and/or constraints upon evolution towards morphological equilibrium. Qualitative judgements concerning estuary behaviour also provide a basis for evaluating the output from quantitative models (i.e. in terms of the direction of predicted change rather than its magnitude). Such insights can be crucial in impact and vulnerability assessment.

Of the various qualitative simulation approaches available a Boolean network formulation is particularly straightforward to implement. It presents few computational challenges, yet has the potential to predict important aspects of estuary system behaviour, including the existence of multiple equilibrium states and the qualitative response of a stable estuary configuration to environmental change (e.g. sea-level rise or sediment supply changes) or interventions (e.g. dredging or coastal protection). Since symbolic Boolean functions can be evaluated directly there is no need for complex graphical or algorithmic system programming and the user has full control over the model logic.

Two distinct classes of software have been reviewed in terms of their suitability for network-based qualitative system modelling and for the implementation of a Boolean model in particular. These include, firstly, packages that allow the user to construct graphical representations of the system of interest before interactively specifying functional linkages and dependencies, and, secondly, numerical/visualisation software that requires the user to implement underlying model algorithms in some form of computer code.

Alternative architectures for an estuary system simulator have been considered. The most promising option appears to be a web-based decision support system comprising a browser-based interface to the EstSim classification of UK estuaries; generic system diagrams and associated behavioural statements; and an option to compute system evolution trajectories from a menu of estuary types and typical management questions. The web-based simulator could usefully be supplemented by standalone open-source 'research-level' code derived from the work presented in this report. Equipped with a simple GUI, the latter would include a library of pre-defined Boolean functions and a set of pre-defined scenarios, but would also allow a more experienced user to specify custom estuary configurations or scenarios through external (open-format) steering files, as well as modify the simulator code to further develop the methodology.

The Boolean network formulation presented by Karunarathna and Reeve (2005) has here been extended and a prototype model implemented in *MATLAB* to demonstrate the capabilities of the approach with respect to the behaviour of idealised estuary systems. Enhancements include the explicit recognition of external forcing/intervention, internal process and morphological variables; spatial sub-division into coastal and outer/inner estuary sub-systems; and the addition of decay terms to condition morphological responses to process forcing. With a minimal initial formalisation of geomorphological understanding, superficially simple estuary systems are shown to exhibit quite complex evolutionary trajectories and both steady state and oscillatory equilibria when initialised at a state of disequilibrium. Encouragingly, the seven generic estuary types defined in the FD2117 EstSim project are discriminated by sensible combinations of forcing factors and morphological components.

Evaluation of the prototype model against a dataset of historic changes in the Ribble estuary yields satisfactory results. The gross evolutionary behaviour of this funnel-shaped estuary is modelled reasonably well by a scheme incorporating outer and inner estuary sub-systems, although abstraction of historic data into a form that is commensurate with the model requires some care. Application of the model to an idealised representation of Southampton Water gives broadly plausible behaviour, although customisation of the generic functions would be needed to resolve detailed aspects of the imposed geology, physical process regime and management history.

The enhanced Boolean network methodology and simulator architecture that has been presented here is extensible both in terms of the variable set and in the number of linked sub-systems. Several important issues have been highlighted concerning the selection of a scale of system abstraction that is commensurate with the assumptions made in the qualitative model, and considerable care is needed in the formalisation of geomorphological knowledge into consistent systems diagrams and model functions. Such issues are not unique to the Boolean approach, although the adoption of single-bit logic with synchronous updating of system states imposes some restrictions on what can be modelled at a given level of abstraction.

There is considerable potential for further work in this field. Areas worthy of particular attention include the improvements in model capability that might accrue from the incorporation of asynchronous updating and more complex sub-system hierarchies; the development of robust protocols and software tools for the coding of logically-consistent Boolean functions; and the operator variance associated with different geomorphological interpretations.

8. THE WAY FORWARD (FUTURE OBJECTIVES)

Objective 5, System Simulation, has provided the system based on work undertaken in Objective 4 using the outcome of the Behavioural Statements Report as the basis for the functions and estuary types.

In order to facilitate this a workshop on EstSim was held at the Marine Research Council on 19 October 2006. The objective of the workshop was to present jointly the results and findings of projects FD2117 and FD2107 and then to have specific project meetings to discuss progress and the way forward. The workshop was attended by all partners of the EstSim project, except for representatives from Plymouth University. Jon French (UCL) described the approach taken for Objective 5 and the key issues discussed included the use of sub-systems to increase resolution of certain components of the system, Boolean definitions, user variance issues, and the simulator capabilities. The inclusion of Southampton Water as a second case study was also agreed. Further discussion took placing on the following objectives including the Management Questions Objective (2).

8.1 Objective 6 Manager - System Interface

Taking the work of the previous Objectives this research element has created a web-based interface allowing managers to explore their estuary types and development using some system tested examples as detailed in the case studies explored in Objective 5. The results of Objectives 5 and 6 will be disseminated at the Defra Conference in July 2007, when an Estuary Research Dissemination Day will also show case the results of the other findings in this project (FD2117) with those of FD2107.

9. REFERENCES

Ahnert, F. (1994) Equilibrium, scale and inheritance in geomorphology. *Geomorphology* 11, 125-40.

Bagley, R.J. and Glass, L. (1996) Counting and classifying attractors in high dimensional dynamical systems. *Journal of Theoretical Biology* 183, 269-84.

Barabási, A-L., Albert, R. and Jeong, H. (1999) Emergence of scaling in random networks. *Science* 286, 510-12.

Bennett, R.J. and Chorley, R.J. (1978) *Environmental systems: philosophy, analysis and control*. London, Methuen, 624pp.

Besa Marchado, V. and Bredeweg, R. (2003) Building qualitative models with HOMER: a study in usability and support. *Proceedings 17th International Workshop on Qualitative Reasoning*. Basilia, Brazil, 39-46.

Bilke, S. and Sjunnesson, F. (2001) Stability of the Kauffman model. *Physical Review* E 65, 016129.

Bossel, H. (1986) *Ecological systems analysis: an introduction to modelling and simulation*. Feldafing, DSE - German Foundation for International Development, 180pp.

Bryant, R.E. (1986) Graph-based algorithms for Boolean function manipulation. *IEEE Transactions in Computing C* 35, 677-91.

Capobianco, M., de Vriend, H., Nicholls, R. and Stive, M.J.F. (1999) Coastal area impact and vulnerability assessment: the point of view of a morphodynamic modeller. *Journal of Coastal Research* 15, 701-16.

Chappell, J. and Woodroffe, C.D. (1994) Macrotidal estuaries. In: R.W.G. carter and C.D. Woodroffe (eds.) *Coastal evolution: late quaternary shoreline morphodynamics*. Cambridge, Cambridge University Press, 187-218.

Chorley, R.J. (1962) *Geomorphology and General Systems Theory*. United States Geological Survey Professional Paper 500-B. US Government Printing Office, Washington D.C.

Cowell, P.J. and Thom, B.G. (1994) Morphodynamics of coastal evolution. In: Carter, R.W.G. and Woodroffe, C.D. (eds.) *Coastal evolution: late quaternary shoreline morphodynamics*. Cambridge, Cambridge University Press, 33-86.

Dalrymple, R.W., Zaitlin, B.A., and Boyd, R. (1992) Estuarine facies model: conceptual basis and stratigraphic implications. *Journal of Sedimentary Petrology* 62, 1130-46.

de Vriend, H.J., Capobianco, M., Chesher, T., de Swart, H.E., Latteux, B. and Stive, M.J.F. (1993) Approaches to long-term modelling of coastal morphology: a review. *Coastal Engineering* 21, 225-69.

Dronkers, J. (1986) Tidal asymmetry and estuarine morphology. *Netherlands Journal of Sea Research* 20, 117-31.

EMPHASYS Consortium (2000) *Modelling estuary morphology and process*. Final Report for MAFF Project FD1401. HR Wallingford, Report TR111.

EstSim Consortium (2004) FD2117 Project Record PR2 *EstSim Behavioural Statements Report*, 127pp.

Forbus, K.D. (1984) Qualitative process theory. *Artificial Intelligence* 24, 85-168.

Forbus, K.D. (1990) The qualitative process engine. In: D.S. Weld and J. de Kleer (eds.) *Readings in qualitative reasoning about physical systems*. San Mateo, CA, Morgan Kaufmann, 220-35.

Ford, A. (1999) *Modeling the environment. An introduction to system dynamics modeling of environmental systems*. Washington, Island Press, 415pp.

Fox, J.J. and Hill, C.C. (2001) From topology to dynamics in biological networks. *Chaos* 11, 809-15.

French, J.R., Reeve, D.E. and Owen, M. (2002) *DEFRA/EA Estuaries Research Programme – Phase 2 Research Plan*. London, DEFRA, 47pp.

Futurecoast (2002) *Predicting Future Coastal Evolution for Shoreline Management Plans*. Defra 2002.

Glass, L. (1975) Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology* 54, 85-107.

Hayes, M.O. (1979) Barrier island morphology as a function of tidal and wave regime. In: S.P. Leatherman (ed.) *Barrier islands from the Gulf of St Lawrence to the Gulf of Mexico*. New York, Academic Press, 1-27.

Hibma, A., Stive, M.J.F., and Wang, Z.B. (2004) Estuarine morphodynamics. *Coastal Engineering* 51, 765-78.

Hopfield, J.J. (1994) Physics, computation, and why biology looks so different. *Journal of Theoretical Biology* 171, 53-60.

Karunarathna, H. and Reeve, D. (2005) FD2117 Project Record PR3 *Mathematical Formulation of Estuary Simulator*, 13pp.

Kaufman, S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* 22, 437-67.

Kaufman, S.A. (1974) The large scale structure and dynamics of gene control circuits: an ensemble approach. *Journal of Theoretical Biology* 44, 167-90.

Kaufman, S.A. (1993) *The origins of order*. Oxford, Oxford University Press.

Kaufman, S.A. (1996) *At home in the universe: the search for laws of self-organization and complexity*. Oxford, Oxford University Press.

Kuipers, B. (1986) Qualitative simulation. *Artificial Intelligence* 29, 289-388.

Kuipers, B. (1994) *Qualitative reasoning: modeling and simulation with incomplete knowledge*. Cambridge, Massachusetts, MIT Press.

Long, A.J., Scaife, R.G. and Edwards, J. (2000) Stratigraphic architecture, relative sea-level, and models of estuary development in southern England: new data from Southampton Water. In: K. Pye and J.R.L. Allen (eds.) *Coastal and estuarine environments: sedimentology, geomorphology and geoarchaeology*. London, Geological Society Special Publication 175, 253-79.

Minato, S. and Arimura, H. (2004) Combinatorial item set analysis based on zero-suppressed BDDs. Hokkaido University, Division of Computer Science, Report Series A, *TCS Technical Report TCS-TRA-04-01*, 11pp.

Mulligan, M. and Wainwright, J. (2004) Modelling and model building. In: Wainwright, J. and Mulligan, M. (eds.) *Environmental modelling: finding simplicity in complexity*. Chichester, Wiley, 7-73.

Nicolis, C. (1982) A Boolean approach to climate dynamics. *Quarterly Journal of the Royal Meteorological Society* 108, 707-15.

O'Brien, M.P. (1969) Equilibrium flow areas of inlets on sandy coasts. *Journal of the Waterways and Harbours Division, American Society of Civil Engineers* 95, 43-52.

Odum, H.T. and Odum, E.P. (2000) *Modelling for all scales: an introduction to system simulation*. London, Academic Press.

Phillips, J.D. (1992) Nonlinear dynamical systems in geomorphology: revolution or evolution. *Geomorphology* 5, 219-29.

Pritchard, D.W. (1967) What is an estuary? Physical viewpoint. In: G.H. Lauff (ed.) *Estuaries*. American Association for the Advancement of Science, Publication 83, 3-5.

Puccia, C.J. and Levens, R. (1985) *Qualitative modelling of complex systems: an introduction to loop analysis and time averaging*. Cambridge MA, Harvard University Press.

Saunders, A. and Ghil, M. (2001) A Boolean delay equation model of ENSO variability. *Physica D* 160, 54-78.

Senge, P.M. (1990) *The fifth discipline: the art and practice of the learning organisation*. London, Century Business.

Seymour, R. (2004) *Notes on network dynamics*. Unpublished research paper produced for ABPmer Ltd. 28pp.

Sipser, M. (2005) *Introduction to the theory of computation*. PWS Publishing, 456pp.

Sklar, F.H., Gopu, K.K., Maxwell, T. and Costanza, R. (1994) Spatially explicit and implicit dynamic simulations of wetland processes. In: W.J. Mitsch (ed.) *Global wetlands: old world and new*. New York, Elsevier, 537-54.

Smith, N. (2000) First steps in formalising qualitative system dynamics. *Proceedings 14th International Workshop on Qualitative Reasoning*. Morelia, Michoacan, Mexico, 138-43.

Thorn, C.E. and Welford, M.R. (1994) The equilibrium concept in geomorphology. *Annals of the Association of American Geographers* 84, 666-96.

Townend, I.H. (2003) Coast and estuary behaviour systems. In: Davis, R.A. (ed.) *Proceedings of the International Conference on Coastal Sediments 2003*. CD-ROM published by World Scientific Publishing Corp. and East Meets West Productions, Corpus Christi, Texas, USA, 13pp.

Townend, I. (2005) An examination of empirical stability relationships for UK estuaries. *Journal of Coastal Research*, 21, 1042-53.

Tullos, D., Neumann, M., and Alvarez Sanchez, J.J. (2004) Development of a qualitative model for investigating benthic community response to anthropogenic activities. *Proceedings 18th International Workshop on Qualitative Reasoning*. Northwestern University, Evanston, Illinois, 7pp.

van der Wal, D., Pye, K., and Neal, A. (2002) Long-term morphological change in the Ribble Estuary, northwest England. *Marine Geology* 189, 249-66.

von Bertalanffy, L. (1968) *General system theory: foundations, development and applications*. New York, George Braziller, 289pp.

Wolstenholme, E.F. (1999) Qualitative vs quantitative modelling: the evolving balance. *Journal of the British Operational Research Society*, 50, 422-9.

Wright, L.D. and Thom, B.G. (1977) Coastal depositional landforms: a morphodynamic approach. *Progress in Physical Geography* 1, 412-59.

Appendix 1. GUI-Based Dynamic System Simulation Software Product Overview

Stella : isee systems, inc

Stella is one of the best-established Graphical User Interface (GUI)-based system modelling tools. Originally developed in 1987, it has evolved into a widely used tool for the modelling of a diverse range of system modelling problems across the natural and social sciences. A recent addition to the range is *NetSim Creator* that allows the creation of web-based interactive simulations that utilise a server-based installation of *Stella*.

Main features highlighted in marketing:

- Intuitive icon-based graphical interface simplifies model building;
- Stock and flow diagrams support the common language of Systems Thinking and provide insight into how systems work ;
- Enhanced stock types enable discrete and continuous processes with support for queues, ovens, and enhanced conveyors;
- Causal Loop Diagrams present overall causal relationships;
- Model equations are automatically generated and made accessible beneath the model layer;
- Built-in functions facilitate mathematical, statistical, and logical operations;
- Arrays simply represent repeated model structure and sub-models support hierarchical model structures ;
- Sensitivity analysis reveals key leverage points and optimal conditions;
- Results presented as graphs, tables, animations, QuickTime movies, and files;
- Dynamic data import/export links to Microsoft® Excel;
- Save as Runtime option creates full-screen, runtime models;
- Multimedia support for graphics, movies, sounds, and text messages based on model conditions.

Standalone models: Yes. Runtime models can be used by users with no access to *Stella* software.

Open source: No, but models are saved in an open-format.

Required expertise: Low-Medium (eliminates requirement for programming skills, but requires some experience and understanding of system simulation principles).

Platform: Windows and MacOS-X systems.

Cost/licensing: single licence US$1899 for commercial users and US$649 for academic users (UK price: December 2005). Volume discounts available.

Web: isee systems : www.iseesystems.com.

ModelMaker : ModelKinetix

ModelMaker is an inexpensive and easy to use Graphical User Interface (GUI)-based system modelling tool. It is designed for scientists who are not necessarily expert mathematicians, programmers or modellers. Using a compartmental method of modelling users can quickly convert a conceptual model into a working mathematical model of a system of interest. The package is intended for application to problems in environmental, biochemical, geological, ecological or pharmacological science.

Main features highlighted in marketing:

- GUI-based system diagram construction;
- Eextensive set of mathematical expressions for formulation of system relationships (including Boolean operations);
- Modelling of continuous and discontinuous functions, stiff systems and stochastic systems;
- 5 different integration methods - Runge-Kutta, Mid-Point, Euler, Bulirsch-Stoer and Gear;
- Optimisation and Monte Carlo simulation capabilities;
- Graphic and tabular output of results, with support for cut and paste into other applications.

Standalone models: No, although model files can be distributed amongst other *ModelMaker* users.

Open source: No, and models appear to be stored in a closed proprietary format. No obvious means of integrating models with other applications.

Required expertise: Low (eliminates requirement for programming skills, and is one of the simpler graphical model building and system simulation packages on the market. However, some understanding of system simulation principles is still needed in order to realise all of its capabilities).

Platform: Windows systems.

Cost/licensing: Single licence £210 for commercial users and £160 for academic users (UK price: December 2005). Printed documentation is available at extra cost. Volume discounts available.

Web: ModelKinetix : www.modelkinetix.com/modelmaker.

Simile : Simulistics Ltd

Simile is a relatively new addition to the range of Graphical User Interface (GUI)-based system modelling tools. Unlike many of its competitors, *Simile* is explicitly intended for the modelling of earth, environmental and life science systems. It uses a logic-based *declarative modelling* approach to represent the interactions with environmental systems in a structured and visually intuitive way.

Main features highlighted in marketing:

- GUI-based system diagram construction;
- Formal systems notation for model conceptualisation;
- Extensive set of mathematical expressions for formulation of system relationships;
- Support for matrix operations;
- Support for hierarchical model-building (including separate execution of sub-models);
- Object-oriented constructs allow extremely concise representation of model structure;
- Completed models can be simulated using compiled C++ code – improving execution speed and allowing the incorporation of *Simile*-developed models into other applications;
- Extensive visualisation and presentation capabilities;
- *Simile* models are stored in an intuitive open (ASCII text) pseudo-code format, which makes the modelling process more transparent to users.

Standalone models:	No pre-packaged runtime viewer provided, but user-developed applications using the open source Tool Command Language and GUI Tool Kit (Tcl/Tk) available at no cost at www.tcl.tk.
Open source:	No, but Simile is very 'open-ended' in its implementation. Models are stored in portable open format and can be readily embedded in other applications developed using C++ or Tcl/Tk.
Required expertise:	Low-Medium (eliminates requirement for programming skills, but nonetheless a powerful package that requires some experience and technical understanding of system simulation principles in order to leverage all of its capabilities).
Platform:	Windows, MacOS-X and x86-linux systems.
Cost/licensing:	Single licence US$495 ($995 for developer licence) for commercial users (UK price: December 2005). Free evaluation version capable of handling models of restricted size.
Web:	Simulistics Ltd : www.simulistics.com.

Vensim : Ventana Systems, Inc

Vensim is a visual modelling tool that allows conceptualisation, simulation and analysis of dynamic systems. It provides a simple and flexible way of building simulation models from causal loop or ‘stock and flow’ system diagrams By connecting words with arrows, relationships among system variables are entered and recorded as causal connections. This information is used by an interactive equation editor to complete the simulation model.

Main features highlighted in marketing:

- GUI-based system diagram construction;
- Equation editor for formulation of system relationships;
- Extensive library of built-in functions;
- Support for optimisation, sensitivity analyses and Monte Carlo simulation;
- Support for external functions (e.g. coded in C++);
- The *Vensim* DLL allows control of *Vensim* from Visual Basic, Delphi or any other programming language;
- Extensive visualisation and presentation capabilities.

Standalone models:	Yes, packaged applications and complied models (C language) can be generated by more expensive versions.
Open source:	No. Compiled model files stored in proprietary binary format.
Required expertise:	Low-Medium (eliminates requirement for programming skills, but nonetheless a powerful package that requires some experience and technical understanding of system simulation principles in order to leverage all of its capabilities).
Platform:	Windows and MacOS-7 (or later) systems.
Cost/licensing:	Single licence US$129 to $1995 (depending on version / functionality) for commercial users (UK price: December 2005). Academic and volume discounts are available. A limited functionality educational version is available as freeware.
Web:	Ventana Systems, Inc : www.vensim.com.

VisSim : Visual Solutions, Inc

VisSim is Graphical User Interface-driven software based around the *VisSim* visual block diagram language for modelling and simulation of complex nonlinear dynamic systems. The software combines an intuitive drag-and-drop block diagram interface with a powerful simulation engine. The visual interface offers a simple method for constructing, modifying, and maintaining complex system models. Its tightly integrated development platform makes it easy to pass freely between model construction, simulation, optimization, and validation, without having to undertake any traditional computer programming.

Main features highlighted in marketing:

- Drag-and-drop block diagram construction;
- 110+ linear and nonlinear blocks (including Boolean functions);
- Toolbox functions for control, electromechanical design, hydraulics, signal processing etc.;
- Euler, trapezoidal, Runge Kutta 2nd and 4th orders, adaptive Bulirsh-Stoer and Runge Kutta 5th order, stiff backward Euler integration algorithms;
- Implicit system solvers;
- Vector and matrix operations;
- Hierarchical models with embedded sub-diagrams;
- 'What-if' scenarios;
- Parameter optimization;
- Synchronous or asynchronous data exchange;
- DLL wizard for custom C, C++, Fortran, and Pascal blocks;
- *MATLAB, MathCAD*, and *Maple* integration.

Standalone models:	Yes. Models can be developed and distributed for use with freeware *VisSim* model viewer.
Open source:	No. *VisSim* model 'scripts' are distributable in open ASCII text format.
Required expertise:	Medium (eliminates requirement for programming skills, but nonetheless a very powerful package that requires experience and technical understanding of system simulation principles in order to leverage all of its capabilities).
Platform:	Windows systems.
Cost/licensing:	Single licence £1995 for commercial users (UK price: December 2005). Discounts available for academic users.
Web:	Visual Solutions, Inc : www.vissim.com.

Extend : Imagine That, Inc

Extend is designed to be a flexible, extendable simulation tool that is primarily intended for the modelling of business organisations and their associated processes. Like other system modelling software, it includes an interactive and graphical architecture that is integrated with a powerful and robust model development environment.

Main features highlighted in marketing:

- GUI-based system ‘drag and drop’ model construction;
- Extensive set of mathematical expressions for formulation of system relationships;
- Support for matrix operations;
- Support for hierarchical model-building;
- Full suite of interprocess tools for communicating with other applications;
- Extensive visualisation and presentation capabilities;
- *Extend* models are stored in open (ASCII text) format.

Standalone models: Yes, using freeware viewer, although this has no print or save functionality. A run time-only version of the software is available at nominal cost.

Open source: No, but *Extend* is ‘open-ended’ in that model ‘block components’ are stored in an open format to allow modification and enhancement. Users can thus alter existing blocks and develop new proprietary components. Linking to code and routines written in external languages (e.g. C, Fortran) is also supported.

Required expertise: Medium (eliminates requirement for programming skills, but nonetheless a powerful package that requires experience and technical understanding of system simulation principles in order to leverage all of its capabilities).

Platform: Windows, MacOS-X systems.

Cost/licensing: single licence US$895 to US$1595 depending on version (December 2005). Complex licensing options, include a run time only licence that costs US$95 and a freeware ‘model player’. Discounts available for academic users.

Web: Imagine That, Inc : www.imaginethatinc.com.

PowerSim : PowerSim Software AS

PowerSim provides a powerful environment for the conceptualisation and modelling of systems, using graphical model building tools that allow for the construction of models according to formal principles of systems theory, and simulation capabilities that allow deductions to be made concerning the behaviour of systems so defined. The software is primarily intended and marketed for use in the modelling of business organisations and industrial process operations. However, it is clearly capable of application to environmental systems.

Main features highlighted in marketing:

- GUI-based system diagram construction;
- Construction of models based on a formal systems dynamics paradigm;
- A graphical modelling language, with a mathematical definition language similar to common spreadsheets;
- Support for logical operations;
- Support for units of measurement;
- Support for hierarchical model-building;
- Powerful presentation capabilities, with easy to use linking capabilities
- Optimisation and sensitivity analysis;
- Data connectivity other applications (e.g. Microsoft *Excel*).

Standalone models:	Yes, using freeware *PowerSim* player.
Open source:	No. However, models can be distributed in an open format such that algorithms employed are accessible to other user of *PowerSim* (or *PowerSim* Player).
Required expertise:	Medium (eliminates requirement for programming skills, but nonetheless a powerful package that requires experience and technical understanding of system simulation principles in order to leverage all of its capabilities).
Platform:	Windows systems.
Cost/licensing:	Single licence £850 to £1750 (depending on version) for commercial users (UK price: December 2005). Academic development pack is £750. Significant discounts are available for multiple purchases.
Web:	Powersim Software AS : www.powersim.com.

GARP3 : Human Computer Science Laboratory, University of Amsterdam

GARP3 was developed under an EU-funded research programme (NatureNet-Redime) led by the Human Computer Science Laboratory at the University of Amsterdam. Essentially, it comprises a workbench for building, running and analysing qualitative models. Models are defined using sets of building blocks that are assembled to describe behavioural properties and scenarios to define the initial conditions for a simulation. The GARP3 methodology is equally applicable to a wide range of system modelling problems, including ecological applications as well as classic engineering and physics examples.

Main features highlighted in marketing:

- GUI-based seamless workbench for construction of qualitative models;
- Construction of models based on qualitative formalisation of expert knowledge;
- No requirement for programming skills;
- Graphical icons for model ingredients;
- Ability to save simulation outcomes.

Standalone models: No.

Open source: Yes. Requires compilation using SWI-Prolog (open source Prolog).

Required expertise: Medium (no requirement for programming skills and easy to use once installed, but requires some prior experience and technical understanding of QR-based system modelling).

Platform: Windows, Linux/Unix (including Mac OS-X) systems.

Cost/licensing: Freeware.

Web: GARP3 Project at the Human Computer Studies (HCS) laboratory, University of Amsterdam: http://hcs.science.uva.nl/QRM/software/.

Appendix 2. Numerical Computation and Visualisation Software Product Overview

MATLAB : The Mathworks, Inc

MATLAB integrates a high-level programming language with an extensive library of functions and other tools for data analysis, visualization, and cross-platform application development. The flexible MATLAB environment incorporates more than 1000 mathematical, statistical, and engineering functions, and interactive graphical capabilities for creating plots, images, surfaces, and volumetric representations. Add-on toolbox algorithms enhance functionality in areas such as signal and image processing, data analysis, and system modelling (notably through the *SIMULINK* toolbox).

Main features highlighted in marketing:

- Powerful high-level language for numerical computation and data analysis;
- Extensible and customisable through user-defined functions;
- A very large library of mathematical, statistical, image processing and other specialised routines (including linear and nonlinear equation solving);
- Support for matrix operations ;
- Powerful visualization capabilities, from 2D plots and image displays to interactive 3D graphics that take advantage of OpenGL hardware acceleration;
- Support for variety of system modelling methodologies within the core *MATLAB* package, as well through optional toolboxes such as *SIMULINK* and *STATEFLOW;*
- Development of standalone applications facilitated by built-in Graphical User Interface (GUI) builder, *GUIDE*, and optional *MATLAB* Compiler;
- *MATLAB Web Server* lets *MATLAB* programmers develop Web-deployable applications. HTML documents serve as a point-and-click GUI for *MATLAB* applications being deployed. Application users are not required to learn *MATLAB* nor need it be not be run locally on client machines.

Standalone models: Yes. Models can be built into standalone applications using an extensive set of user-interface tools and the (optional) *MATLAB* Compiler.

Open source: *MATLAB* itself is 'open-ended' in that its functionality is extensible and user-customisable, although underlying kernel is proprietary closed-source software. Script codes are distributable in open ASCII text format.

Required expertise: Medium - high (research level software with a fairly high threshold of knowledge and steep learning curve for novice user, although user interface is very sophisticated).

Platform: Windows and Unix-like (including MacOS-X) and linux systems.

Cost/licensing: Single floating network licence £2995 for commercial users (UK price: December 2005). Add-on toolboxes cost extra (~£500 each). Substantial discounts are available for academic users.

Web: The Mathworks : www.mathworks.co.uk.

IDL : Research Systems, Inc (RSI)

IDL integrates a high-level programming language with an extensive library of functions and other tools for data analysis, visualization, and cross-platform application development. The feature set is particularly strong with respect to image-based analysis and visualisation and data mining. The software environment caters for interactive analysis and display as well as large-scale commercial programming projects. *IDL* does not include a suite of tools explicitly intended for system simulation, but simulation models can easily be coded, distributed and executed in the form of scripts.

Main features highlighted in marketing:

- High-level language for numerical computation and data analysis;
- Extensible and customisable through user-defined functions;
- A rich library of mathematical, statistical, image processing and other specialised routines;
- Support for matrix operations;
- Powerful visualization capabilities, from 2D plots and image displays to interactive 3D graphics designed to take advantage of OpenGL hardware acceleration;
- Graphical User Interface (GUI) toolkit for *IDL* ‘application’ development;
- Support for standalone applications distributed in proprietary binary format can be run with the free *IDL Virtual Machine* or with a ‘runtime’ license of *IDL*.

Standalone models: Yes, using free runtime-only model ‘viewer’.

Open source: *IDL* itself is ‘open-ended’ in that its functionality is extensible and user-customisable, although underlying kernel is proprietary closed-source software. Runtime-only models distributed in closed proprietary format, although script codes can be separately provided in open ASCII text format.

Required expertise: Medium - high (research level software with a fairly high threshold of knowledge and steep learning curve for novice user, although user interface is very sophisticated).

Platform: Windows and Unix-like (including MacOS-X) and linux systems.

cost/licensing: Single floating network licence £2995 for commercial users (UK price: December 2005). Substantial discounts available for academic users.

Web: RSI – IDL : www.rsinc.com/idl.

Mathematica : Wolfram Research, Inc

Mathematica seamlessly integrates a numeric and symbolic computational engine, graphics system, programming language, documentation system, and advanced connectivity to other applications. It is particularly capable in the area of symbolic mathematics and equation solving.

Main features highlighted in marketing:

- Powerful and flexible high-level language for symbolic and numerical computation and data analysis;
- Support for complex symbolic calculations that involve thousands or millions of terms;
- Support for matrix operations;
- Large library of mathematical, statistical, and visualisation functions;
- Powerful visualization capabilities, from 2D plots and image displays to interactive 3D graphics;
- Ability to embed *Mathematica* functionality into applications developed using high-level languages such as C or Visual Basic, and develop interactive web applications that run on a *webMathematica* server.

Standalone models:	Yes. Models can be built into standalone applications using an extensive set of user-interface tools and the (optional) MATLAB Compiler.
Open source:	*Mathematica* itself is 'open-ended' in that its functionality is extensible and user-customisable, although underlying kernel is proprietary closed-source software. Script codes are distributable in open ASCII text format.
Required expertise:	Medium - high (research level software with a fairly high threshold of knowledge and steep learning curve for novice user, although user interface is sophisticated).
Platform:	Windows and Unix-like (including MacOS-X) and linux systems.
Cost/licensing:	single *Mathematica Professional* licence £1625 for commercial users (UK price: December 2005). Add-on packages and web-enabled services cost extra. Substantial discounts available for academic users.
Web:	Wolfram Research : www.wolfram.com.

MathCad : Mathsoft

Mathcad is an integrated software environment for performing and communicating maths-related work. Like *Mathematica,* it is particularly capable in the area of symbolic mathematics and equation solving, and as an interactive presentation tool.

Main features highlighted in marketing:

- Numeric operators perform summations, products, derivatives, integrals and Boolean operations;
- Symbolics simplify, differentiate, integrate, and transform expressions algebraically;
- Vectors and matrices manipulate arrays and perform various linear algebra operations, such as finding eigenvalues and eigenvectors, and looking up values in arrays;
- Ability to generate random numbers or histograms, fit data to built-in and general functions, interpolate data, and build probability distribution models;
- Ability to generate random numbers or histograms, fit data to built-in and general functions, interpolate data, and build probability distribution models;
- 2D and 3D visualization capabilities;
- 'Live' maths technology allows use of mathematical expressions using standard mathematical notation - but with the added ability to recalculate, view and publish results easily, including to the Web;
- Ability to embed *Mathcad* functionality into applications developed using high-level languages such as C++, Java or Visual Basic, with connectivity with *VisSim*, *MATLAB* and Microsoft *Excel* also featured.

Standalone models: No, although Mathcad has the ability to interface with external applications such as Microsoft *Excel* and *MATLAB*, as well as applications developed using common high-level programming languages.

Open source: *Mathcad* itself is extensible and user-customisable, although underlying kernel is proprietary closed-source software.

Required expertise: Medium - high (research level software with a fairly high threshold of knowledge and steep learning curve for novice user, although user interface is sophisticated).

Platform: Windows systems.

Cost/licensing: Single *Mathcad* licence £758 for commercial users (UK price: June 2006). Add-on packages cost extra. Discounts of around 50% or more available for academic users.

Web: Adept Scientific : http://www.mathsoft.com/.

GNU Octave : John Eaton and others (published via Free Software Foundation)

GNU Octave is freely redistributable software that integrates a high-level interactive language, primarily intended for numerical computations, with 2D and 3D visualisation capability. The software is command line driven, and the command language used is that is mostly compatible with MATLAB (although only a subset of MATLAB functionality is currently implemented). The software does not include tools explicitly intended for simulation, but simulation models can easily be coded, distributed and executed in the form of scripts.

Main features highlighted in marketing:

- High-level (MATLAB compatible) language for numerical computations;
- Extensible and customisable through user-defined functions;
- Wide range of elementary functions (including support for Boolean operations);
- Specialist functions for linear and nonlinear equation solving;
- Support for matrix operations.

Standalone models: No – unlike Matlab or IDL, there is no compiler or suite of embedded tools for application development.

Open source: Yes (both application software and any user-developed model scripts).

Required expertise: High (research level software with a high threshold of knowledge and steep learning curve for novice user).

Platform: Windows and Unix-like systems, but requires compilation with freely available GNU C++ compiler.

Cost/licensing: Free to download and can be freely redistributed under terms of GNU public licence.

Web: GNU Octave: www.octave.org
Free Software Foundation: www.gnu.org.

Appendix 3. Main Morphological Components for EstSim Generic Estuary Types as Defined in FD2117 PR2

This basic morphological variable set is extended in the present report to include additional coastal and fluvial system components as well as a spatial division into outer and inner estuary sub-systems.

Morphological variables:

Barrier beach	BB
Sand spit	SP
Ebb and flood deltas	ED, FD
Linear banks	LB
Channels	CC
Sand flat	SF
Mud flat	MF
Saltmarsh	SM
Dunes	DU
Cliff	CF
Rock platform	RP
River	RI

Morphological variables (potential):

1. Fjord: BB, SP, CC, SF, CF, RP, RI

2. Fjard: BB, SP, CC, SF, MF, SM, CF, RP, RI

3. Ria: BB, SP, CC, SF, MF, SM, CF, RI

4. Spit –enclosed: BB, SP, ED, FD, CC, SF, MF, SM, DU, CF, RI

5. Funnel-shaped: BB, LB, CC, SF, MF, SM, DU, CF, RI

6. Embayment: BB, LB, CC, SF, MF, SM, DU

7. Tidal inlet: BB, SP, ED, FD, CC, SF, MF, SM, DU, CF

Appendix 4. Example of Steering File Format for Prototype Simulator

```
%1 Steering file for ESTSIM boolean system simulator
%2 Run:  ESTSIM WP5.2 testing
%3 Notes: This file represents a stable tidal inlet
%4 Notes:
%5 Format: first line contains number of variables; subsequent lines give
variable names,
%6 Format: followed (optionally) by number of scenarios and the initial
variable states
50
ocean_waves
longshore_power
littoral_sand
marine_sand
marine_mud
macrotidal
mesotidal
microtidal
updrift_beach
spit
inlet
rock_basin
fluvial_channel
downdrift_beach
dunes
wind
prism
accomm_space
ebb_delta
flood_delta
linear_banks
bedrock
rock_sill
outer_rock_platform
outer_flood_dominance
outer_ebb_dominance
outer_subtidal_sands
outer_estuary_swell
outer_estuarywaves
outer_sandflat
outer_mudflat
outer_marsh_low
outer_marsh_high
outer_floodplain
outer_cliff
inner_rockplatform
inner_flood_dominance
inner_ebb_dominance
inner_subtidal_deposits
inner_estuarywaves
inner_sandflat
inner_mudflat
inner_marsh_low
inner_marsh_high
inner_floodplain
inner_cliff
bayhead_delta
river_discharge
river_sand
river_mud
```

```
1
1  ocean_waves
1  longshore_power
1  littoral_sand
1  marine_sand
1  marine_mud
0  macrotidal
1  mesotidal
0  microtidal
1  updrift_beach
1  spit
1  inlet
0  rock_basin
0  fluvial_channel
1  downdrift_beach
1  dunes
1  wind
0  prism
0  accomm_space
1  ebb_delta
1  flood_delta
0  linear_banks
0  bedrock
0  rock_sill
0  outer_rock_platform
0  outer_flood_dominance
1  outer_ebb_dominance
1  outer_subtidal_sands
0  outer_estuary_swell
0  outer_estuarywaves
0  outer_sandflat
1  outer_mudflat
1  outer_marsh_low
1  outer_marsh_high
0  outer_floodplain
0  outer_cliff
0  inner_rockplatform
0  inner_flood_dominance
1  inner_ebb_dominance
1  inner_subtidal_deposits
0  inner_estuarywaves
1  inner_sandflat
1  inner_mudflat
1  inner_marsh_low
1  inner_marsh_high
0  inner_floodplain
0  inner_cliff
0  bayhead_delta
0  river_discharge
0  river_sand
0  river_mud
```

Ergon House
Horseferry Road
London SW1P 2AL

www.defra.gov.uk

defra
Department for Environment
Food and Rural Affairs