EM story

1994. Agent-oriented modelling (Bryan) —

programming — essence is in prescribing and interpreting the behaviour of reliable state-changing devices.

CS requires — rigorous computer-based modelling methods that take account of active real-world situation and enable us to combine principles of abstract mathematical modelling with observation and experiment.

EM requirements — many of the most relevant issues cannot be addressed by conventional mathematical modelling - a radically different framework for modelling behaviour is required.

Our advantage — derived from the idea of systematically refining our system model by using experiment to identify more and more precisely which factors serve to determine the behaviour.

— the power to model interactive experience distinguishes our approach from most other computational frameworks many of which are based solely upon abstraction for describing circumscribed behaviour.

Underpinning model — 2 complementary kinds of activity: - i) refinement of agent models in the light of experiment, ii) commitment to experience of faith in how agents can be expected to operate.

1999 EM and foundations of AI (Bryan) -

contrast to EM — physical artefacts are needed to communicate contents.

an EM model — is empirically established, experientially mediated, agent-centred pragmatic, only accounts for changes of state in the system (e.g. digital logic).

EM proposes — to be a modelling in an env development paradigm.

observable — a characteristic of my environment to which I can attribute an identity. An observable returns a current value.

state — of the world for me, as of now, is represented by a collection of observables with particular values.

dependency — patterns are fundamental to the perception and recognition of observables and determine when they can be deemed to have an identity as an object.

definitive notation — is used to formulate a family of definitions whose structure is to be very similar to the syntactic network of dependencies behind the calls of the [illegible]

definitive scripts – a basis for representing constraints

What EM is not – modelling behaviour is not a primitive concept. In EM an behaviour is a sophisticated ~~abstraction~~ abstraction that invokes the attribution of an identity to a pattern of state transitions.

EM suggests – a broader philosophical framework ~~which~~ within which these are associated with circumscribed and reliable, recurring patterns of experience.

1997 EM for EL – (Beynon)

EM is concerned with – i) developing system models in ways that allow flexible adaptation, extension and re-use even by users who are not computer specialists, ii) developing techniques that allow machine independent specification of software, iii) using computers in ways that address the needs of the artist/craftsman, iv) supporting an open-ended rather than a closed-world engineering culture.

EM predicated on theory – that cognition and learning are fundamentally concerned with a process of constructing phenomena in terms of experiential regularities

EM offers – situated modelling with a potential for open interaction that involves creative observation and interpretation of the situation to which the model refers.

in EM model development – has an open flexible character.

EM, 2 principal techniques – for analysis and representation: i) definitive representation of task and ii) observation oriented analysis of expertise

EM provides – a radical view that agrees with observation in relation to the development of understanding.

adequate account of EM – has to accommodate both the empirical and rational elements in human understanding, and explicate the relationship between the two.

ability to observe unexpected – is an essential part of the behaviour of an intelligent agent – the fact that we cannot represent experiential knowledge formally at present should not divert us from its fundamental significance

EM stuff (2).

1992 - New paths for programming (Beynon)

definition of agent - i) a person or thing that acts or exerts power.
ii) any natural force acting on matter; iii) one authorised to delegate or transact business for another.

1995 - Worlds before and beyond words - (Beynon/Russ/McCarty)

Conventional paradigms - are not well-suited for constructing models by empirical methods. Amongst widely used tools only spreadsheets are well-adapted to imitating real world state as it is captured through observation and experiment

antecedents of EM - are found in the techniques for communicating designs and conceptions of the real-world through using physical objects whose features are imitative rather than symbolic

most significant feature of EM framework - is that the model and situation reside within the realm of experience of a single agent, so that it is possible to introduce aspects of the model into the situation

The essence of EM - found where the experimenter acts as the sole state-changing agent

The virtue of an EM framework - is that the identification of observables is provisional in nature, and is subject to revision in the light of further experience.

EM is multi-agent - (or perceived to be) when the changes to observables are not all attributable to actions on the part of the experimenter.

distinguish from conventional - the emphasis on modelling state as directly experienced rather than behaviours as circumscribed.

1999 - Computer mediated Communication - (Beynon/Sun)

distributed EM - a client/server architecture. Interaction between modellers is shaped by the interactive mode of communication established by the server.

Current technology - as computer-mediated communication technology matures it promises to enfranchise human interaction as rich in concurrency, situation & embodiment as we are accustomed to find in our most intimate interaction with our environment

EM not - to be viewed as implementing an abstract mathematical model.

EM called - because its interpretation is shaped by open-ended interaction with both the model and (at least in principle) also with its referent.

Significance of EM - it offers principles to account for how we experience represents another : principles that can be used both to validate such a representation and to create one.

agent in EM - refers to any family of observables that can be construed as associated with a coherent identity

EM places emphasis - on the identification of dependencies as a basis for analysing and representing experience

proper framework for ISM construction - is itself based around families of observed dependencies and concurrently acting agents.

The role of an ISM - is not in the first instance to prescribe a pattern of activity to suit a particular function, but to track an evolving state of mind.

distributed EM environment - is engineered so that the state-changing actions that each participant is able to carry out can be interactively shaped to reflect their capabilities.

fundamental aspect of distributed EM (dome server) - is mediating between the perceptions of the participants. This has practical implications for the ways in which investigation of the model is conducted.

2001 EM of products ~~Fisher/Beynon~~ (Beynon/Fisher)

Central to EM - an emphasis on the power of the computer to represent states, in particular states which are easily interpretable.

An EM model - metaphorically represents a particular state of the system under study

EM based on - three fundamental concepts; observables, dependency and agency

observable - used to refer to relevant elements of the referent

dependency - a relationship amongst observables that expresses (indivisible) expectation about how the values of observables are linked in changes.

agent - has privileges for action to change the values of these observables.

In an EM artefact - any new aspects can be added to the artefact at any stage of during the modelling process without the need to

EM Mag (5)

... revise the whole artefact or even re-start the process

EM distinguished — from conventional modelling approaches where the inputs and outputs must be determined in advance, before starting the construction of the model, in order to prescribe the required behaviour

The EM environment — can give the impression of animation of the artefact through automatic re-definition of observables and consequent visualisation of several and consecutive representative states of an artefact.

EM enables — the user to build an artefact that is always open to revision and extension, providing the user with a high degree of interaction with the model.

EM environment — offers a high level of interaction with the model allowing the modeller to do constant revision in order to obtain the best representation of the product.

EM features — immediate visualisation of artefact modification and model animation are very important for increasing the knowledge of the modeller about the product.

2001 — EM for the conceptual design and review of engineering products — (Cornall... / Beck...)

(AOM 94-98)

EM is — an approach to constructing models that emphasises interaction and experimentation with the model in all aspects of the design activity

major advantage of an EM model — is it's flexibility

EM based on — three key concepts, namely observables, agency and dependency, but these are conceived initially as personal constructs drawn from our own experience of the domain that we are modelling

An ISM — models the relationship between a real world situation and an observer of that situation.

The power — of our development method derives from the idea of systematically refining our system model by using experimentation to identify more and more precisely what factors serve to determine the behaviour.

Purpose - to model immediate experience distinguishing an approach from most conventional frameworks that are based solely upon abstraction for describing constructed behaviour.

199x - ISM for Information System Developers (Beynon, Cornelius, Rasmequan)

An ISM - provides an environment within which the human investigator can explore the relationships between observables and the patterns of behaviour associated with a system component, with particular reference to its external real-world semantics.

an agent - refers broadly to any component of the system that can be responsible for changing state.

EM system - a framework for containing discrete systems in which the basic abstractions are observables, dependencies between observables, and agents that may through changing observables affect dependencies.

an observable - is some feature of a system to which a value or state can be attributed in a system state. Empirical procedures and conventions are involved in identifying a particular observable and assigning it a value.

an agent - is a family of observables whose joint and observed system states is correlated in time, that is typically deemed responsible for particular changes to observables within the system.

a dependency - is a relationship between observables that persists in the view of a particular agent.

to Em - context a situation involves 2 correlated activities: i) developing the analysts explanatory account of the situation in the modeller's mind; ii) constructing an ISM to mirror the observed response to experimental exploratory interaction.

Construction an ISM - involves creating a model to enable a pattern of observables, dependencies and agents that faithfully reflects observation of the relevant.

Cont. today (?)

The ISM - serves as a metaphorical representation for the system, consistent with the modeller's construal.

In an ISM - the situation is represented by a script of definitions

Change of state in an ISM - conceptually should reflect as far as possible the expectation of the modeller.

In the EM framework - the modeller as agent is the archetype for all agents within the system.

From an EM perspective - the events that are enabled in a particular context are determined by the agents and observables that are present, and by the privileges the agents have.

Experimental activities associated with EM - are precisely directed at determining which events can occur in a particular context.

Most significant feature of ISM use - is that their semantics is moulded through situated interaction, rather than abstract and preconceived.

openness of the ISM in EM - can be exploited in debugging and testing, with potential for flexible redesign and fault-tolerance.

—

(Oct 1998) - ISM's for System Development (Beynon/Cartwright/Rungrattanaubol/Sun)

An ISM - is a computer-based environment constructed through situated modelling activity.

An ISM - unlike a closed-world computer model with a fixed interface, an ISM is always open to elaboration and unconstrained exploratory interaction.

States within the ISM - metaphorically represent pertinent situations from the application domain and possible transitions between states are explicitly constructed so as to be consistent with the developer's construal of a system in terms of agents, observables and dependencies.

The ISM is elaborated - by the developer in an open-ended manner as understanding of the domain and the system is acquired.

ISM's - give particular support to the early stages of system development.

An ISM - does not necessarily represent goal-oriented knowledge, it represents subjective knowledge, represents knowledge about interaction in particular contexts.

an ISM – a paradox one may be useful in communication between one modeller and another, it reflects the perception and experience of a particular person of a particular matter is a real

In an ISM – the computer is used to generate a direct metaphorical representation a problem-system, and to animate the modeller's expectations about what latent state-changing actions are available, and what immediate consequences result.

All changes of state – while the ISM is at the discretion of the modeller, they may be delegated to automatic agents if desired, and can be executed without inference in order to simulate particular observe behaviour

an ISM – implicitly attaches some measure of plausibility to the potential responses of the system.

ISM – a synonym for "model created using SM principles and tools"

an observable – a feature of a system to which a value or state can be attributed in a system state.

an agent – a family of observables whose precise and observe in system state is correlated in time, that is typically deemed to be responsible for particular changes to observables within the system

a dependency – is a relationship between observables that pertains in the view of a particular agent.

2 Conversations aspects – to the shared modelling process, i) describing the abstract explanatory account of the situation that the modeller has in mind, ii) Conversations on ISM to initiate the observed response to experimental and exploratory interaction.

LSD – account should not be mistaken for a formal model of system behaviour

The ISM – is constructed as a metaphorical representation for the system that is consistent with the modeller's construal.

purpose of the ISM – is to enable the modeller to assess the quality of their explanatory account by exploring the implications of particular patterns of agency and interaction

The ISM – is a model of the relationship between a situation and an observer, so that a modification of the model can either reflect a development in the situation or a new realisation on the part of the observer.

ISM construction – is essentially concerned with the creative tension between 'what can be observed' and 'what can be explained'.

Em slides ⑤

EM prescribes - no general method for system development.
EM is not - a silver bullet for software technology that the so-called "scientific method" is a recipe for creating scientific theories.
EM offers - is a framework of principles and tools that promises to deliver greater conceptual integrity across the entire development process.

1993 - Agent-oriented modelling for a bilingual simulation - (Forbes/Beynon/Yung)

Our modelling method - corresponds closely to intuitions about the behaviour of real-world systems.
Our modelling method - attributes changes in system state to the action of agents
— are well suited to implementing feedback.

1986 - ARCA: A notation for displaying and manipulating combinatorial objects (Beynon)

ARCA - is a programming notation which was originally designed with the interactive display and manipulation of "Cayley Diagrams".
The design of ARCA - suggests a number of directions for further work. Definitive notations may be useful in other interactive applications, not necessarily concerned with graphics.

1982 - The Interpretation of States: A New foundation for computation (Beynon/Ross)

Computation is viewed as interpretive task-change that can be associated with physical systems of different kinds.

1987 - Definitive principles for interactive graphics (Beynon)

PostScript - has been designed as a popular graphics system illustrating definitive principles

1986 - DONXD: A (line-drawing system based on definitive principles (Beynon Angier Ellis Gay Hunt

1987 - Implementing a definitive notation for interactive graphics - (Beynon/Yung)

1998 - Modelling stuff in mind and machine (Bergen)

Observables - there are 3 kinds, visual, scientific and conceptual.
EM agent - is a different context from other procedural or functional roles
in EM - it is possible to introduce triggered procedures that are invoked when
the values of variables are changed.
advantage - the constructive nature of EM allows experimentation with a wide
variety of objectives.
EM - resembles the experimental method in science.
the EM process - as described here, centres on animation that serves a useful
purpose in demonstrating the concept principle, but does not
directly provide an executable version of the algorithm for use
on a large input array
1994 - Debugging from an agent-based perspective (Bergen/A. Coninsx/Y

A definitive script - can represent the state of a computer model at a suitable
level of abstraction, subject only to the creativity of the
person that uses formulae.
LSD - is used for specifying how each agent is privileged to observe variables
and to change the state of the system through redefining variables.
oracle - a variable that can be observed by an agent.
handle - a variable whose value can be conditionally redefined.
protocol - the set of privileges of an agent.
ADM - the operational interpretation of an LSD specification is expressed using the
the use of Matlab scripts - emphatically close to activity that is not yet understood
in the conventional meaning of the term.

1996 - Artificial geometry for design in an EM context (Bergen/Y/Coninsx R

ramifications for EM - are broad and urgent many directions for potential future development
1) using the computer as a physical artifact to explore phenomena in which
the set of possible states and transitions is unknown. Discovery
for the purpose of development of means through exploration & experiment
2) including in existing representations to allow users different aspects
of the design product in the collaborative assessment process.

Cm slides (6)

EM - is proposed as an approach to constructing systems in such a way as to deliberately encourage openness.

development of a model - involves several loosely sequential phases, i) the identification of observables, ii) the selection of observables to make viewpoints, iii) the identification of entities and the accompaniment of constraints governing possible states of the world, iv) the development of scenarios for state change and environmental interaction.

recognition - of a variable in a definitive script is a means of exposing monitoring of user actions, including voluntary choice of action, stimulus-response patterns, and acts of God.

main tasks in building computer-based artefacts - i) agent-oriented modelling of simulated, ii) imprecise specification of scenarios, iii) interface specification of scenarios, iv) dependency maintenance, v) version management.

the LSD account - of a phenomenon involves identifying the observables associated with the interaction between agents, and expressing the potential actions of each agent in terms of them.

LSD artefacts - have proved useful not only as a route to conceptual system simulation but also as a means of analysing role-level interactions in the construction of an artefact.

- the ADM - is used to synthesise an artefact to represent a system of interacting LSD agents. It supplies a framework within which the current state of a system is specified by a family of definitive scripts that integrate the viewpoints of all the component agents.

principle of EM - is to maintain dependencies between variables and to act as an interface between definitive scripts at a high-level of abstraction and the low-level procedural utilities that handle window management and graphical display.

1998 - Explanatory models of human-computer interaction (Beynon, Sim, Winder, Rajan)

objective of EM - can be seen as the construction of a computer environment that metaphorically represents a slice of the external world.

openness of the EM process - is indicated by the fact that a single EM model can (in principle) incorporate such diverse representations of the current state of room as a floorplan image, a 2D room layout, and a circuit diagram with dynamic annotations to represent the status of the electrical wiring.

EM process - combines 2 fundamental principles: observation-and-agent-oriented analysis and definitive (definition-based) representation of state.

EM is concerned with the empirical process through which knowledge and understanding are acquired, an EM model is in itself essentially incomplete, and must be constructed by interaction.

Constructing an EM model - for a referent is associated with learning through interaction.

EM process - revolves around 3 fundamental concepts: observables, dependencies and agents.

observable - is used to refer to elements of a situation that have identity through time.

dependency - is a relationship amongst observables that expresses expectations about how the values of observables are ordinarily linked in change.

agent - is an observable (generally identified with a collection of more primitive observables) that is deemed to be responsible for changes of state within the situation.

product of EM process - is a causal account for the observed regularities and autonomous behaviour of the referent.

An EM model - is based upon a family of observables, dependencies and agents.

In an EM model - observables and dependencies are represented by a family of definitions.

The agent roles of observables and dependencies represent the current state of the referent.

LSD - developed to specify the characteristics of agents in respect of interactions with their environment.

(LSD) sensor - observable associated with an agent.

effector - observables are deemed to influence the agent.

handle - observable that are conditionally under its control.

dependency - the dependencies between observables that characterise the interaction of the agent with its environment.

EM models can be exploited in 2 ways - i) to attach an interface to an EM model to form computer programs with conventional functionality, ii) can be used in conjunction with interfaces that invite the user to engage with the empirical process both its construction

'25 years (!)

The EM model - can serve a novel role in communication, embodying insight in an explicit form suitable for reuse through interaction.

1994 - Empirical modelling of requirements (Beynon / Russ)

to provide system behaviour - already we need to have identified as the relevant i) observables of the system, ii) agents that can be active in the system, iii) stimulus-response patterns (protocols of agents), iv) dependencies between system parameters, v) timing characteristics of agent interactions and reactions.

our modelling process - has 4 mutually complementary, and inseparable features, i) it is state based (the current state of the real-world context is always represented), ii) it is agent-oriented (identification of the state-changing agents active in the system determines the model structure), iii) it is definition-based (i.e. uses specialised like definitions to express the dependencies perceived between observations in the world), iv) the fundamental entities are observables. These are associated with agents, they define the state, and correspond directly to the variables which are the concerns of the definitions.

dependencies - are modelled by unidirectional constraints, and expressed using systems of definitions, or definitive scripts.

modelling the behaviour of a system of agents - Step 1 is the description of those observables that are bound to an agent (state variable), more it is conditionally privileged to change (handle), and those to which it responds (trade).

2 further features of the modelling process - i) it is a remarkably flexible process allowing on-line re-definition during animation of the model. ii) the physical faithfulness that we strive for in the component models lends an explanatory power to the overall model.

1999 - Cultivating requirements in a situated requirements engineering process (Sim / Cheri Russ Beynon)

through experiment and observation - the modeller confronts an external situation in terms of the primitive concepts of EM: observables, dependencies, agents and agencies, and correspondingly confronts a computer model that metaphorically exhibits similar patterns of behaviour.

EM - is a means of confronting knowledge in an experiential rather than a declarative form. The modeller's insight is expressed as a coherence between expectations in the mind and the experiments that can be performed on the computer-based artefact and/or in the external environment.

Computational states in EDEN - are suited to the broad view of agency referred above:
state changes can be initiated either by human agents or by automatic procedures
in EDEN model - of a reflect is referred to as an Interactive Situation Model.
The activities involved in this evolution of EDEN - i) the identification of agents,
ii) the Competition for roles for these agents commensurate to their Standard
skills, iii) the apportioning of responsibilities for particular phases within a
given transaction, iv) the refinement and formalization of their precise
observables and protocols.

Distributed nature of EDEN - enables us to separate the viewpoints of the agents
in the model, and to contrast these with an external interpretation.

1985 - Parallel Computation in Definitive models (Reynolds / Slade / Young)

LSD - an approach aimed to modelling the interaction between agents in a
concurrent system using definitive principles.

1987 - Definitive programming for parallelism (Reynolds)

Definitive programming - is based upon the use of definitions that establish functional
relationships between variable values.

Definitive programming - is neither purely procedural nor purely declarative.
- reverts to the traditional view of a computation as a
sequence of transitions from state to state.
- is set apart from functional programming through its non-
declarative aspects.

LSD notation - an agent-oriented notation for representing concurrent systems that is
also based upon definitive principles.

1988 - Definitions for modelling and simulating Concurrent systems (Reynolds / Nance / Sloan)

LSD - is an activity-oriented approach based upon a new language for the description of
concurrent systems of interacting agents, is founded upon a novel "definitive
(definition-based) programming" paradigm and has several unusual characteristics.
LSD - also includes features for modelling the implications of action.

Concept of an LSD agent – has its origins in a simple problem for non-computer interaction.

In LSD – a concurrent system is modelled by a family of agents, each having its characteristic oracle, state and derived variables.

motivating idea – is that an LSD specification identifies those aspects of the system behaviour that depend upon the interrelated capabilities and perceptions of its participating agents.

primary role of an LSD model – is to describe the independent asynchronous behaviour of concurrently acting sequential agents, each responding to its perceived view of the system.

an LSD specification – makes explicit the preconditions for an agent to perform an action. also makes it possible to distinguish between synchronisation through perception and synchronisation through interaction assumptions

LSD guides – an agent-oriented view of interactive and concurrent systems.

Paper – Protocol verification in concurrent system software development (Bowman, Slade, Jin

Agent-oriented programming – involves identifying the agents in a system and describing the interactions between them in terms that are easy to relate to the requirements of the application.

Agents are of 2 kinds, i) reactive – can perform autonomous actions to change the system state, ii) passive – able to record state information but cannot perform actions.

Agent-oriented programming – introduces more powerful abstractions to represent communication and synchronised interaction between agents

An LSD specification – correlates possible actions of an agent with its perceived states of the system and provides the basis for simulating the behaviour that exhibit definitive state-transition computational models.

An agent-oriented model – is new suited to the way in which a designer initially conceives the requirements specification for

1989 - Software Construction using definitions: An illustrative example (Pages / Notes / Runs / Slides / Years)

LSD - is a medium for describing the capabilities and perception of agents within a system.

primary concept - "of LSD is the agent: for each agent there are 3 kinds of variable, i) oracles - represent values that influence the behavior of the agent and that are subject to change beyond the agent's control, ii) states - that are controllable under the control of the agent, iii) derivates - record relationships that form part of the context for the agent's actions.

An LSD specification - provides a simple model of agent privileges and protocols.
- in the first instance is to be interpreted as expressing the powers of agents.

Within the ADM - state information is encoded using definitive systems, but there is also provision for automatic changes of state according to a pre-defined protocol

1981 - Definitive principles of the specification of software (Pages / Runs / Slides / Years / Years)

In the ADM - definitive programming provides an unusual parallel programming paradigm that is sufficiently versatile for the implementation of CAD software

Central abstraction of definitive programming - is the functional relationship between variables appropriate to a particular action in the state

The aim of the "definitive programming project" - is to formulate a general-purpose programming paradigm based upon abstractly representing agent actions using definitive systems.

Both LSD and the ADM - have in common a framework of grouped actions.

In LSD - the grouped actions make up the protocol for a sequentially acting agent, and the events express enabling conditions for actions to be met by the agent's view of the external system.

Both LSD and ADM - represent an approach to modelling concurrency "from the study of relative and functional dependencies between events, rather than from the representation of absolute time intervals."

EM Story (4)

1997 - Formal specification from an observation-oriented perspective. (Beynon, Rungrattanaubol, Sun & Wright)

EM - approach allow exploration of the state, and the effect of dependencies between observables, compared to a formal specification which fixes the important features of the system under development.

EM - puts its fundamental emphasis on observation and experiment.

Character of EM activity - can best be motivated by referring to the way in which experimental scientists have developed their understanding of phenomena.

EM - can be viewed as contributing to the science of constructing in 2 complementary ways, i) it offers principles that can be used to frame explanatory accounts, ii) it introduces new practical techniques and tools for constructing constructs.

EM provides - skill construing phenomena to identifying observables, dependency and agency

observable - a feature of a situation that is perceived to have identity and integrity.

dependency - a relationship amongst observables that expresses expectations about how the values of observables are indivisibly linked in change.

agent - an observable that is deemed to be responsible for changes of state within the situation.

EM tools - enable incremental and interactive extension, refinement and revision of a computer model.

use of EM tools - also makes it possible to introduce automatic agents into models

1998 - EM: A new approach to understanding requirements (Beynon, Sun)

EM - involves computer-supported modelling with an emphasis on three...

observable - a characteristic of the modelled environment to which an identity can be attributed

dependency - an empirically established relationship between observables

agent - an instigator of change to observables & dependencies

agency - represents a permission to access an observable

EM - is a powerful form of interactive modelling. Through experiment & observation the modeller construes an external situation in terms of the ... within the course of EM, and concurrently constructs a computer model that metaphorically exhibits similar ... observables, dependencies and agency.

EM - is a means of constructing knowledge in an experiential rather than a deductive fashion.

EM activities - are carried out with respect to an external situation even though in practice this situation can be imaginary rather than concrete

EM can support - collaborative interaction in real situated and distributed contexts.

EM - is oriented towards a collaborative relationship, where the interaction among all participants is situated, distributed and pre-ended.

1998 Enabling technologies for EM in graphics (Allbeck/Badler/Cerezo/Y. Yu

EM - is distinguished from conventional computer-based modelling techniques by its emphasis on the experiential aspects of modelling.

EM methods - are distinguished by the essential character and nature of the artifacts they generate, rather than by any particular functionality that can be desired for them.

EM puts the emphasis - upon representing states and not behaviours, upon metaphorical rather than propositional representation, and upon experience & observation rather than theorem and inference.

central concepts behind EM - definitive (definition-based) representation of state, and agent-oriented analysis and representation of state-transitions.

primary aim of the EM-process is to construct a computer-based model (the Artifact) in which, by the use of suitable ~~metaphors~~ metaphors for interaction & visualisation the human agents who interact with the model can comprehend the behaviour of agents within the Referent.

in practice, the EM process typically serves to identify valid communicative protocol for interaction with definitive variables.

essential quality of EM artifacts - is the integration of 2 kinds of knowledge, i) tacit associated with empirical insights that are yet to be fully explored and confirmed as valid, and ii) acquired knowledge about agency and the environments in which they interact.

EM aims - to establish a more direct relationship between application & machine, without the artificial intermediate constraints that a conventional computer programming perspective introduces.

(10) EM 11-05

The DAM = (Definitive Assembly Maintenance) machine is a low-level definition manager that maintains dependencies between words in RAM store.

1997 Empirical modelling principles in application development for the disabled (Beynon, Cartwright)

In developing an LSD specification - the behaviour of the system is construed in terms of agents and the features of the system state to which these agents react and through which they speak.

In EM - the preliminary design activity is captured using computer models based on imitating the values of the key observables within a system, typically by using visualisation

Model construction in the EM framework - is a human-centred activity in which the emphasis is upon human-computer co-operation, and the central theme is the exploration and interactive adaptation of a computer-based artefact that reflects evolving insight into the system requirements

1996 Spreadsheets and programming (Gehring)

Definitive notation - use the spreadsheet principle of dependency maintenance to model real world systems.

EDEN - is a hybrid programming language supporting both definitive & procedural language elements.

a definitive system - is very open, in that any variable can be related to any other
in EM - the principles of interaction & experimentation are very useful.

1996 Higher-order constructs for Interactive Graphics and Design in a Definitive Programming Framework (Gehring, P Yung, RYA Cartwright, Beynon)

In our EM process - we aspire to comprehend the states of a phenomenon in terms of higher-order observables.

typical example of such observables - i) indivisible relationships represented by definitive scripts
ii) constraints represented by logical relationships
iii) associations of observables represented by data structures.

Comparison with conventional modelling environments - Sketchpad offers the modeller an ... for ... of ... ... & ...

1995 Empirical Modelling principles for Cognitive Artefacts (Beynon / R. Cartwright)

a cognitive artefact – is an object or environment that is contrived for interaction that imitates interaction with some other system.
observable – encompasses any discernible feature of a phenomenon that can be ascribed a value by a well-defined experimental procedure or convention for interpretation.
Concept of state – presumes a viewpoint and a snapshot.
a viewpoint – is associated with an agent who can in general both observe and experiment with artefact and referent.
essence of an artefact – is that it should imitate interaction with its referent.

1995 Computer-assisted jigsaw construction: A case-study in EM (Beynon / Sidebotham / ...)

essential focus of EM – reflects the dictionary definition of empirical: "that which is based upon observation and experiment".
EM – addresses the process of transforming from experiential to theoretical perspective.
basic principle of EM – is the construction of conceptual representations of real-world phenomena.
a definitive state – (generalised) differs from states in ordinary procedural or object languages in that relationships between variables are implicit in the definitions of the variables.

x-ref check.
1994 An agent-oriented framework for concurrent engineering (Adzhiev / Beynon / A. Cartwright / Y.P. Yung)

In deriving an ADM simulation – of a concurrent system from LSD specification of its constituent agents, we impose a regime for the execution of each agent's privileges to change system state in response to changes in its environment.
In the ADM – the modeller has exceptional privileges to intervene & reprogram dynamically.

1994 A Computer-aided script generator for CAD (Y.P. Yung / Beynon).

application of definitive principle – to interactive graphics was first introduced by Brian Wyvill in 1975.

(1) 251 steps.

An agent - has certain privileges to perceive its environment and certain privileges to influence its environment.

LSD - is a notation we have developed for specifying agent privileges.

In a LSD specification of an agent we specify the oracle variables which the agent can perceive, the handle variables which the agent can modify, the state variables whose existence depends on the presence of the agent and the derivate variables which are defined in terms of other variables. There is also a protocol section to specify the actions the agent can perform when the state it perceives meets certain conditions.

2 computational models for the implementation of LSD specification - ADM & SOS

SOS - is more procedural in style.

1994 Interactive geometric modelling based on R-functions. (Adzhiev/Rvachev/Pasko)
an agent-oriented approach

Our approach differs fundamentally from an object-oriented paradigm in that -
i) message passing is only one mode of agent interaction
ii) direct action of one agent upon another is possible.
iii) single action typically effect state changes in several agents in one indivisible transition.

An LSD specification - is aimed towards understanding the interaction between agents in a system by applying principles intimately connected with experience and observation.

State variable - agent owns.

Oracle variables - to which an agent responds.

Handle variables - that are controllable under its control.

Derivate variables - represents indivisibly coupled stimulus-response relation.

Each agent has a protocol - that specifies its privileges to change the system state subject to certain enabling conditions being met.

The complete LSD specification - provides a clear and concise description both of the interaction between user & modelling system and of the main processes taking place through interaction between the components of the modelling system.

Our LSD specification - can be interpreted operationally using finite state to represent computational state.

1994 A new computer-based tool for conceptual design (Beynon/Beynon/... [illegible])

conceptual design - a negotiation between what we believe and what ...
our modelling method - exploits a new form of agent-oriented modelling based
on representing agent actions by redefinition of variables in the
context of a script of definitions.
important features of our approach include - i) principled modelling that is intimately
connected with observation & experiment in the physical world, ii) effective way to
represent synchronisation and concurrent action, iii) potential for integration with
existing CAD methods and tools.
the distinction between our modelling methods & traditional approaches to computer
programming reflects the profound difference between theorising and experimentation.
perceived state - a particular set of measurements of observables such as might be the
~~[illegible]~~ in a single experimental reading
primitive state-transition - we refer to the changes to these measurements that would
result from a single "atomic" action on the part of the modeller.
a definitive script - as a model of an experimental observation represents a primitive
hypothesis about how the world behaves.
definitive script - experimental hypotheses about primitive behaviour in highly
constrained circumstances
- are appropriate when the relationship between observables is still
subject to exploration.

1994 A computational model for multi-agent interaction in concurrent engineering (Adzhiev/Beynon/Cartwright/Yung)

In our approach - agent privileges are specified in a special notation LSD.
In an LSD specification -
oracle - a variable that can be observed by an agent
handle - a variable whose value can be conditionally redefined.
protocol - the set of privileges of an agent.
derivate - values that are derived from others by fixed rules of [illegible] or constraints

(12) EM story

1993 Programming as Human-Computer Interaction (Beaver)

1993 A new paradigm for parallelism in Engineering Application (Beaver)

In effect, the LSD specification - identifies the experimental observables that would have to be made in order to explain the behaviour of the agent in isolation

epsode - observations to which an agent reports
handle - thing, which it can conditionally change.

1993 Agent oriented modelling for engineering design (Beaver/A. Catharsis)

Our approach - is based on certain fundamental abstractions for describing state-transition models for complex interactive systems

each agent - has certain perceptions of state and certain privileges to change state.

epsode - value perceived by the agent
handles - how it can conditionally redefine
derivables - indivisible relationships between observed values.
state of the agent - the same variable may be referenced in different ways by different agents. All such references are viewed as a ~~handle for another, the~~ ~~such references are viewed as~~ authentic value of the variable and is bound to a particular agent and accordingly defines part of the state of that agent.
protocol - the enabling conditions that must be met before an agent can perform a state-changing action.

ADM - was developed as a medium for animation from an LSD specification. One of definitive scripts has many advantages in implementation. i) SCRIPS expresses indivisible propagation of state-changes through a complex system. Makes it easy to link components that are developed independently. ii) Scripts also make the dependencies between variable values explicit, so that the re-evaluation involved in changing state can be efficiently identified.

1993 Modelling a social system using definitive principles (Beynon / Lee)

Developing definitive programs - involves 3 complementary activities:
i) agent-oriented analysis - takes account of how each agent observes changes of state in other agents and conditionally privileges to change the state of other agents.
ii) representing experimental knowledge - involves constructing set of definitions in which variables correspond to experimental observations and the definitions encode expectations about how changes to observations are correlated with state changing actions are performed.

Definitive script - represent particular states of knowledge that can be revised/modified to reflect new observations

Most significant advantage of a definitive approach - we can revise program interpretation that were not perceived when setting up the model.

1994 Agent-oriented modelling and simulation for distributed event systems (Beynon / Yu-Hsin)

Our state-based models can be developed incrementally to correlate model behaviour with experiment/observation
- can be interpreted by the designer in terms of agent actions in the world
- can be used for simulating and analysing system behaviour

Agent-based programming - involves identifying the agents in a system and describing the interactions between them in terms that are easy to relate to the requirements of the problem

Definitive programming - describes a state-transition model of the system in terms of series of definitions and redefinitions

We differ from OOP - no concept of information hiding is present
- message passing is only one of the ways in which agents interact

modes of interaction outside the scope of OOP - direct action of one agent upon another
- single actions that effect state changes in several objects are indivisible transactions

An LSD specification for an agent describes -
- the aspects of the system state to which it can respond (cycle)
- those aspects it can conditionally change (handles)
- circumstances under which state-changing actions can be performed (protocol)
- also includes definitions that can express the different ways in which agent actions are to be interpreted in state-transition terms. (definitions)

En slaan (13)

Our experience shows that our approach towards the process of model design / simulation -

i) it leads to the construction of state-based models that can be validated against experiment.

ii) it creates a model that relates the global system behaviour to the characteristics of participating agents.

iii) it supports incremental development of a complex model as well as easy recovery during backtracking.

iv) it enables rich modes of interaction in simulation.

1992 Scientific simulation: Experiments and observation (Beynon / Yi-Yun / A. Cai / Mayo)

The agent-oriented nature of our programming framework - we presume here are state-changing agents other than the experimenter.

a phenomenon - is conceived of as associated with a set of states, as defined by the comprehensive ob of simultaneous observation

AMA notation - is oriented towards displaying and maintaining enumerational diagrams

Scan notation - is designed for screen layout.

The use of LSD - represents one way in which we have successfully combined definitive principles for representing state with a view to general-purpose programming

1992 Enhancing Interaction in Computer-Aided conceptual design (A. Cai / Beynon)

Definitive scripts - are a powerful way of representing data dependency.

1992 Agent-oriented modelling for a Vehicle Cruise Control System (Beynon / Bridges / Yi-Yun)

The state variables - are the variables owned by an agent,

The audit variable - represent ~~variables~~ value to which an agent responds.

The driver variables - represent stimulus-response relationships that are indivisibly coupled

An LSD specification - is useful to document the way in which the behaviour of a system depends upon the characteristics and interrelationships of its components

1991 Programming principles for visualisation in Mathematical Research (Beynon/Y.P.Yu/Ahmad/Bird)

A definitive notation – is a simple prescriptive medium in which scientific reasoning about underlying spreadsheet style formulae

The EDEN interpreter – automatically monitors data dependence and updates variable values efficiently through selective re-evaluation

A script – can be used to represent knowledge about functional relationships between data that persist throughout transition from one state to another.

1990 Environments for mathematical research: a project report (Beynon)

ARCA – was the first definitive notation to be developed.

Definitive state representations – derive their power from the simple principle illustrated in the spreadsheet. A set of definitions can be used to represent knowledge about functional relationships between data that persist throughout transition from one state to another.

1990 Programming paradigms and the semantics of Geometric Systems (Beynon/Rutt/Y.P.Yu/A.Cormack)

Abstract

1990 Progressive modelling: new concepts and techniques (Beynon/Rutt/Y.P.Yu)

The state represented by the script – has a quite fundamental and essential relationship to the interpretation of the associated image.

In enabling a meaning to computer representation of objects, 3 interrelated ingredients have a fundamental role – i) a state-based representation, ii) agents and their privileges to change state, iii) an exact correspondence between state changes in the application and the model.

In a DST (definition-based state transition) Model – the values of a system of variables are defined either explicitly or implicitly in terms of other variables in a non-cyclic fashion

DST representations – are the means by which we represent the admissible transformations of system of characteristic variables.

an LSA specification – records the variables to which the agent can respond (oracle), those variables that it can conditionally change (state), and those variables whose values are defined by functional relationships (derivate).

(14) EM Hewitt

3 primary concerns are significant here - i) individual actions have to be accurately characterised as described, ii) the manner in which the agents in the system act + interact in relation to their view of the system must be captured and represented. iii) the way in which the concurrent action of agents is constrained by environmental factors must be taken into account".

1990 Definitive interfaces as a Virtualisation Mechanism (Beynon / Y. P. Yung)

a "script" - is used to represent state and can be dynamically modified during execution to reflect the prevailing relationship between the screen state and the internal state.

a definitive notation - is a simple programming language in which a program is a sequence of concise declarations and definitions.

a set of definitions - serves as an effective computational device for representing the propagation of change.

LSD - does provide a suitable framework in which to model the salient characteristics of agents.

1990 Definitive specification of concurrent systems (Beynon / Norris / Orr / Slade)

The behaviour of a complex system - is typically first conceived by the designer in terms of the interaction between "agents".

definitive expressions - ~~also often~~ are well-suited to expressing the effects of agent actions as conceived by the system designer.

In LSD - a concurrent system is modelled by a family of agents, each having its characteristic scalar, state and private variables.

The primary role of an LSD specification - is to describe the interaction between agents in a concurrent system in terms of their abilities to perform actions

1990 Parallelism in a definitive programming framework (Beynon)

1989 Definitive programming as a framework for design (Beynon)

Definitive principles - are particularly ~~very~~ suited to expressing constraints in which the interaction between user and computer

A definitive system - is a family of variables such that the value of each variable is either specified explicitly, or is defined by a formula in terms of constants and other variables.

In the implementation of definitive programming - the transition from state to state is entirely under the control of the user.

Definitive programming has many potential merits for design support:

1) it provide a state-based programming paradigm.

2) it allows agent privileges and actions to be explicitly modelled.

3) it supports rich techniques for data representation and presentation.

Agent protocols - are generally implicit in design.

1991 The Development and use of variables in Mathematics and Computer Science (Bergman/Rees)

Procedural use of variables - serves as a direct way to capture state.

procedural methods - reflect the manner in which state is manipulated in a conventional computer.

Declarative programming methods - have evolved in response to the difficulties of representing relationships in a procedural framework.

Purefunctional programming system - most effectively eliminate state.

It appears - that more powerful methods of dealing with sets of valuations as states and state transitions are required, and that it is necessary to consider when, and by what agents, state changes are enabled.

1988 A definitive programming approach to the implementation of CAD software (Bergman/Gaston)

A definitive notation - is specified by an underlying algebra comprising a set Δ of data types and a family Σ of operators that have the form of pure functions mapping between the data types.

The use of a pure definitive notation - for user-computer dialogue puts the emphasis upon representing the current state of the interaction by means of a system of definitions.

The EDEN interpreter - has built-in support for a definitive notation based upon list-processing, but can also be programmed to perform limited procedural actions that may be synchronised with changes in the dialogue state using triggering mechanism similar to those used in OO systems.

(15/EM sheets.

EDEN - makes it possible to link complex procedural actions and intricate system of definitions: a very powerful mixed programming paradigm, but one that can also prove difficult to analyse

1988 Representing design knowledge in a definitive programming framework (Bennett/A. Cox)

Pure definitive notation - provides a framework within which it is easy to describe relationships between entities within the same semantic category

The use of a definitive programming paradigm - leads to a much closer integration of issues concerned with the user-interface and the abstract application.

1989 Evaluating definitive principles for interactive graphics (Beaven)

definitive programming - provides a much broader perspective within which to consider the use of "imperative features".

demonstrate that a user would ideally wish to make a design system one that is -

i) relationships should be easily perceptible, and conveniently modified.

ii) it should be possible to record partial information about relationships conveniently

iii) relationships should be expressible at many levels of abstraction.

iv) it should be possible to accommodate a temporary failure to meet constraints, be compensate at a later date, and conveniently make consequent changes retrospectively.

1981 Parallel Computation in definitive models (Bagrodia/Slade/Yung Yu)

Conceptual advantages of a definitive interface - stem from the fact that it makes explicit both the parameters that the user can change and the consequences of such changes.

In effect, EDEN provides - a practical programming tool that makes it easily possible to link complex procedural actions and intricate systems of definitions: a mixed programming paradigm that has proved to be potentially very powerful

1988 The 20th Rank Xerox (Y. Wilks).

1988 Definitions for modelling and simulating concurrent systems (Pagan/Wainwright)

LSD - is founded upon a novel "definitive (definition-based) programming" paradigm
A distinctive feature - is that the perceptions and capabilities of agents are explicitly modelled.
This is important because :- (i) it becomes possible in principle to consider the implications of altering the perceptions of agents. (ii) it is possible to distinguish between synchronisation based upon an agent's perception and synchronisation based upon implicit assumptions about the speed with which protocols are executed.
LSD provides - an agent-oriented view of interactive and concurrent systems.

1988 Implementing a definitive notation for interactive graphics (Beynon/Wang)

The DoNaLD notation - is intended for the interactive display and manipulation of planar diagrams comprising points and lines.
a potential problem with DoNaLD - is that it provides methods of programming that may be powerful but can be obscure and difficult to analyse.

1988 Definitive principles for interactive graphics (Beynon)

In a definitive programming notation - the emphasis is on interaction.
ARCA - is a more sophisticated notation than DoNaLD, and might have been intended as an experiment in software design rather than a practical programming medium.
a definitive notation - which enables the user to specify new operators and data types in principle supplies a more general framework.

1987 Comparison of SDL and LSD (Beynon/Warren)

The LSD notation - has a small set of primitive constructs and concepts
A definitive notation - is a natural generalisation of a spreadsheet in which the variables may be of many different sorts.
The LSD notation - was conceived of as a convenient medium for representing processes as a first step towards describing the complex relationships involved in communication and synchronisation.

(10 in sets)

1987 The LSD notation for communicating systems. (Beynon)

The LSD notation - is intended for the specification and description of communicating systems of processes acting concurrently.

In interaction using a definitive notation - the state of the dialogue is represented by a combination of variables with explicit values and variables which are implicitly defined by formulae.

Simplest examples of state variables - are variables which are conditionally under a person's control.

In developing the LSD description - the first step is to identify the states, derivates and oracles for the principal processes.

The behaviour of the system - is defined by the possible transitions to and state of the same form which can occur through the action of the live process instances.

The LSD notation was originally conceived - in connection with semantic models for the CCITT standard language SDL.

1986 Definitions for programming (Beynon)

For a definitive programming system - the current state is determined by what defns of functions, or equations, have taken place in the course of a user dialogue.

For a procedural programming system - the current state is determined by what variables have been declared, what procedures have been defined, and the current values of variables.

For a functional programming system - the current state is determined by what functions have been defined, and

For a logic programming system - the current state is determined by what logical variables have been defined and what equational constraints have been put upon them by predicates.

Interaction - naturally involves ideas which are most conveniently viewed procedurally in that significant interaction must possibly affect a "change of state".

The significant aspect of a definitive notation - is that it allows the state of a dialogue to be modelled in a way which is conceptually simple but which has advantages over alternative approaches.

within a definitive system - there is no useful concept of computation state, but only an ambient state in which certain variables have been given explicit values, and other variables have an implicit value specified by an associated definitive formula.

1985 Definitive notations for interactive (Reynar)

Definitive notations - are conceived as a notation for dialogues over some limited universe of discourse.
Definitive notations - are conceived as a framework within which some professional oriented notations for interactive use can perhaps be developed.

Earlier 2000 Strategic Decision Support system: An experience-based approach. (Ramesh / Rao / Ram)

EM - the emphasis on observation and experience gave rise to some principles of EM - have philosophical roots that have much in common with the work of William James.
EM approach - offers a distinct aspect in modelling
in EM - we are focusing - initially in the state of a model or domain, rather than behaviour we wish to produce (one of the reasons for the flexibility)
EM - is a collection of principles, tools and techniques for an alternative approach to system development.
EM offers - an unusual environment where the way in which a system is developed begins with a very pre-ended analysis of a domain.
EM offers - an environment where it is reasonable to expect that any useful components from one model can be integrated with another model.
ESD - is an informal notation in which an initial account of a domain identifies what seem to be the relevant events and observations.
In EM - the influences between modeller and artefact are 2-way, the modeller expects to gain new insights from the developing model and these will affect future development.
EM - generates spreadsheets in a way that has significant practical implications

The development of EM artefacts - is an open-ended exploratory activity

An EM model - more general than a spreadsheet in the following 3 major respects

i) perception, ii) underlying algebra, iii) agency

The concept of agency in EM - enables dependency relationships to be manipulated

~~EM approach -~~ concurrently by several agents, both human + automated

EM approach - is very far-reaching in its scope.

The Temposcope: a cognitive instrument for the idealist traveller (Beynon / Wardman / Wongsaroj / Russ)

The computer based Interactive Situation Model (ISM) - is intended to represent situations and not concerned with specific process, in mind like a computer program.

In broad terms, the EM approach - to computer-based modelling can be viewed as generalising principles illustrated in the spreadsheet so as to create computer-based artefacts whose semantics resembles that of the physical models that an engineer or experimental scientist constructs.

The key concepts of EM - are the observable, dependency and agent.

The central idea of EM - is the construction of an ISM.

An ISM - is expressed as an open environment

The particular state of an ISM - is determined by a family of definitions that resemble the network of definitions of cells that lies behind the spreadsheet interface

Script of an ISM - serves as the basis for an exceedingly rich state-transition model.

Our ISM-based approach - has more in common with modelling cognitive artefacts than implementing efficient automatic routines.

The underlying assumption in EM - is that our knowledge is incomplete and imperfect, and liable to be revised and improved in the future.

Most significant aspects of an ISM - are that it is open-ended, and, in so far as it represents experience of the reality, these are consistent with experiment and observation.

An ISM - does not easily explain mindless interaction

2000 Cognitive artefacts for decision support (Ramaprasad, Rao)

EM - because our approach is based on theories and exp
a major motivation for EM - is to complement systems, or rule-base
models with models which offer the user-experience of interac
that directly corresponds with real world experiences
ISM's - because we draw attention to the unconstrained interaction
possible with the models, and the fact that they have a
situated quality in contrast with generic, abstract models
typical of mathematical modelling methods
An ISM - is a computer-based artefact that is used to represent
shared-as-experienced
An ISM - especially in the early stages of construction, is a private
and subjective model with the here-and-now character of
a particular situation well known to the modeller.
The principles of EM - are at close to the way humans operate in our eve
can be viewed as a generalisation of procedures
Our ISM - has given different character in being developed to
change in the experience and the situation of use

2000 The use of ISM's for the development of business solutions (Pegan Ramaprasad Rao)

The central activity of EM - is the building of computer-based artefact
that permit to the user the key elements of a domain in a way
that the user will readily recognise them
The model of an ISM - is motivated by thinking about state
An ISM - is a computer-based artefact that is used to represent
shared-as experienced
An overview - any aspect of the current state of the object to which
an inquiry and identity can be attributed
Interaction with the ISM - involves a characteristic state-based
orientation towards the referent
Models - demand human interaction throughout the whole activity
EM models - do not only include an explicit representation of the
procedural knowledge of the domain expert or modeller but can also

(18 Ensktog

The principal function of the ISM - is to convey interactions that involve other agents in terms that make this projection explicit.

An ISM - is a means to represent the modeller's - in general partial and evolving - knowledge of state and agency in the external world.

Empirical modelling for business process re-engineering: An experience based approach. (Chen/Russ/Beynon)

It is essential within EM to take account of the wider context of a desired 'system' in terms of the purposes, people and other resources which will form the environment of the system.

An ISM - is open to experiment in much the same way that its real world referent is open to experiment.

We can devise changes in an ISM - introduce new factors, and have direct experience of the results in any way we choose at the time.

The word 'situated' in ISM - refers to the fact that the model is rooted in a concrete context that steers the modeller's expectation and interpretation.

LSD classifies observables as - those which an agent can act as stimuli for actions (oracles), those that can be conditionally redefined (handles), ones whose existence is tied to that of the agent itself (states), the relationships between observables expressing the interaction between the agent and its environment (derivates), and the privileges of agents for state-changing actions (protocols).

A fundamental difference between EM and conventional modelling - in the way that the modeller interacts with the state of the model.

With an EM model - participants can (in principle) have a close and continuous engagement with each other.

It is a feature of EM - that the conventional phases of system development (specification, design, implementation, testing etc.) tend to be conflated and are continuously elaborated during the evolution of our models.

What we are aspiring to do in EM - is to introduce a new, powerful electronic modelling medium for the shared construction of artefacts that can faithfully and flexibly embody existing, and planned, real world systems.

EM - is a human-centred approach which can have tools supporting distributed working with sophisticated modes of communication

The EM approach - differs in a significant respect - the boundary of system is not preconceived but grows with the understanding of the modeller

The EM approach - pays more serious attention to the true character of real-world observation.

The observational and interactional context for each agent - oracles are the observables it can refer to, handles are those it can conditionally change, and the protocol that connects these.

view 1 agent - an agent has unexplored potential to affect system state

view 2 agent - an agent may be construed as reliably following some particular pattern of stimulus-response within the system

view 3 agent - each agent enacts a pattern of stimulus-response interaction that can be entirely circumscribed and predicted

EM promotes the idea - that the concept of agency is only meaningful in relation to the development of understanding from view 1 to view 3 perspective.

The LSD account - can be viewed as identifying and classifying the observables that capture the modeller's current understanding of the specification.

The distributed version of EDEN - enables us to separate the viewpoints of the agents in the model, and to complement these with an external observer's interpretation

EM - computer-based models can be built in a way similar to that in which human make conceptual models of phenomena.