ON DEFINING THE GRAND CHALLENGES OF SYSTEMS DEVELOPMENT AND LOOKING TO BIOLOGY FOR INSPIRATION

Neil McBride, School of Computing, De Montfort University

Abstract

This paper discusses some of the grand challenges which face systems developers. It suggests that new methods will be needed to encompass evolvability, integratability, complexity, socio-technical alignment, reusability and individuality. It reviews the design characteristics that new commercial information system may need and suggests that at its heart this is a design science problem. Using a metaphor-driven approach to design, it then turns to biology for inspiration and uses concepts from molecular biology to generate ideas which may lead to solutions to the grand challenges posed at the start of the paper. The paper concludes that there are possibilities for pursuing new ideas by exploring biology as a source of system design ideas, not only at the detailed implementation level, but also at a higher design and analysis level. While some of the resulting ideas may be speculative, it is suggested no idea is incapable of improving our knowledge.

Keywords: Biologically-Inspired Computing, Grand Challenges, Design Science.

1 INTRODUCTION

Systems development practitioners face a wide range of challenges in the 21st century as a result of technical and social transformations in the use of information and communication technology (ICT) in the social and business world. The effect of the World Wide Web has been to create global information systems, to move the computing interface out of the business administrator's office and into general society and to increase the complexity and variety of interactions with information systems. The variety of applications has increased dramatically and moved away from a focus on transaction systems to a varied fauna of information systems dealing with information and communication needs. This has resulted in a painful awareness of the inadequacies of current systems development methods and a search for new approaches which is at the heart of the study of design science in information systems. Globalisation, married to rapidly changing markets and short product lifecycles, has resulted in a need for faster systems development and in the adoption of agile methods and rapid development methods whose approach is more analogous to jazz improvisation and the evolving of design over time than to the construction of architectural artefact, monoliths designed once and meant to be immortal.

These business demands for more malleable and rapidly produced information systems create a need for constructs, models, methods and instantiations based on different paradigms. The traditional philosophies behind information system development will need to be challenged and new frameworks for designers developed. The top down approach, which derives a complete specification for a system may have to be replaced by bottom-up approaches which develop small, evolvable elements of systems. Fixed specifications like grand plans for an entire housing development, may be replaced by evolvable specifications which address small immediate problems and allow ill-defined hooks for hanging new design on at a later stage. This is, in part the basis of service-oriented architectures. However, in such a dynamic system development paradigm, classical object-oriented approaches such as the Unified Modelling Language are inadequate for expressing dynamic requirements. New approaches which marry the requirements for service definitions with the design of core components are needed. A search for new approaches to systems analysis and design is now required.

However, system design approaches which attempt to remove complexity and create absolute certainty may have to be abandoned. Uncertainty, unresolved complexity and incompleteness will have to be lived with and accepted.

This will require system developers to take an interdisciplinary approach and be prepared to draw on ideas and theories from other disciplines. This paper suggests biological sciences as a candidate discipline and considers how some design concepts in molecular biology might transfer to systems development and aid the development of new methods and new ways of constructing ICT artefacts. The paper develops an understanding of the challenges facing system designers, the nature of the systems that might emerge from meeting these challenges is discussed and the role of biologically-inspired computing introduced. The paper then explores the value of three phenomena from molecular biology in system design. This exploratory paper concludes by discussing the agenda for extended research in the use of biological metaphors to develop ideas, tools and IT artefacts.

2 GRAND CHALLENGES IN SYSTEMS DESIGN

While the grand challenges in systems design are generated by range of environmental, technical and social changes, the actual characteristics of the changes needed in the process of design and nature of the IT artefact can be articulated under a number of generic headings. The following outlines six challenges which arise from the nature of the change and the inadequacies of current systems.

2.1 Evolvability

Evolvability concerns the ability of an artefact or system to make appropriate changes in response to the demands of the environment. Constant change to information systems are generated by the evolving business demand. However, both at an implementation and design level, such change is difficult to achieve. Systems developers talk about freezing requirements. Such a process inevitably results in a drift of the system from the business as the business changes. Even worse, the freezing of requirements may result in the alignment of the business with the system such that business processes are frozen in time, encoded in legacy systems, such that the business cannot change. Changes may be of various kinds (Table 1)

Type of Change	Example
Rule / process	We don't do X any more. Process Y involves a change to step A.
Data definition	The patient no is fourteen digits now. We no longer store Z
Organisational	Process X is now done by department Y Departments A and B have merged.
Data	We no longer sell to Y
Scope	Function Y has been stopped
User ability change	This is Mr X he doesn't know about Function Y
Function merger	Department Y of Company A and Department Y of Company B are merged.

Table1. Environmental stimuli which require an evolving response from a system.

Evolvability should be possible at both the design and implementation level. System designers need to ensure that the IT artefacts can adapt to changes of the range suggested in Table 1. Changes in design should be possible in implemented systems. Evolvability will require changes in the systems development life cycle such that continuous structural changes, perhaps triggered by the system are possible. It may require an emergent design approach in which much of the system design evolves during and after implementation. System adaptation is a vital component of commercial systems. Different approaches to design will be needed which are more able to leave open ends and allow new functionality to accrete in an existing system.

2.2 Integratability

System design needs to consider issues of boundary definition and communication between systems as much as communication within systems. Systems integration has always been a key issue within organisations. It is now essential to address integration of systems between organisations. IT artefacts should be aware of where their boundaries are. Information systems should be adaptable enough to communicate with new systems. They should know what is part of the system and what is part of a foreign system. They should be able to differentiate between self and non-self. Design methods are needed which concentrate on the system interface with users and other systems as much as the internal data and process structure. Design should take into account the adaptive responses necessary. Adaptive responses to changing stimuli should be able to trigger adaptation in design. The focus of systems development methods needs to move away from concern with internal processes, data and interaction in the system towards the interfaces between systems and a model in which the prime analysis focus is on the system as a set of relationships with other systems and users. Systems development is often primarily an exercise in systems integration. Currently, both computer systems and the design methods have little awareness of boundaries.

2.3 Complexity

In most systems development exercises the prime objective is to conquer complexity and reduce the system to a number of simple rules. The nature of large systems on the web and within organisations is that the level of complexity is too great and does not yield to the reductionist philosophy of systems analysis. Complex interactions which are irreducible and give rise to emergent behaviour are much more likely to occur in systems. Alternative approaches to system design which live with complexity are required. These approaches should deal with complex interactions, being able to represent them appropriately. Future analysis methods may have to accept than areas of the system will remain complex.

2.4 Socio-technical alignment

The alignment of the technical representation of the system within an IT artefact with the social reality of the system remains a problem. Attempts to close the gap between systems analysis models and the messy real-life problems have proved difficult. A computer representation currently requires exact definition and the removal of uncertainty and ambiguity. Systems need to align with business behaviour and user behaviour. Designers of information system need to understand group behaviour, community behaviour and community interaction. In particular web-based system will be affected by community interactions and will affect communities. Understanding of group and individual behaviour issues needs to be incorporated into the system design method.

2.5 Reusability

Reusability remains a thorny issue in systems development. Systems developers are still building system in which a majority of the design is created from scratch. Reusability may be limited to parts of programs at a low level. Patterns themselves do not constitute reuse since they provide guidelines and frameworks within which the designer builds new artefacts, rather than being ready-to-use. Reuse is more likely in implementation at the component level. Design reuse needs a revival in attention. Common domains of business process need to be recognised and generic processes established.

2.6 Individuality

Current commercial systems are difficult to adapt to individual user preferences. They tend to fix business processes and limit organisational development. Adaptive systems are needed which while working at an organisational level with core business processes, can also adapt to the work flow, processes and needs of individual users. They should enable the expression of individuality within defined business constraints. Design approaches need to allow for specifying major individual preferences. Such design concerns are a major part of taking a service-oriented view of information systems, tailored to individual needs and not seeing the IT artefact as a mass produced product.

A key problem here is the need to balance diversity with commonality. Current systems development approaches drive at commonality, looking to fix on one solution, to be used by all, and encoded in the IT artefact. However, the drive towards corporate systems, based on one design, inhibits diversity.

Diversity and individuality are necessary both to create innovation and derive competitive advantage. Current systems development methods are monolithic and based on engineering and production metaphors or models rather than artistic and service metaphors. Hence there is a need for service-oriented approaches to systems design.

3 THE NATURE OF NEW IT ARTIFACTS

The outcome of addressing these challenges in system development will be new types of information systems and IT artefacts in which more flexible and adaptable and are not constrained by the system development approach used. New IT artefacts will be:

- Designed from the bottom-up, extendable and plastic;
- Based on coherent building blocks of design. These building blocks will define core ideas, concepts and units of design which can be applied in a number of situations;
- Interface focused, able to identify and adapt to new systems and even self-adapt to develop new interfaces to new systems, aware of their boundaries, able to differentiate between self and non-self;
- Able to identify and respond in their design to changing stimuli;
- Not dependent on a master plan but more a result of accreted design;
- Designed with the assumption that they are distributed;
- Designed with in-built flexibility to adapt to different users, different business contexts and different interfaced systems;
- Automatically reconfigurable to fit new environments;
- Based on networks of interaction which give rise to emergent behaviour which is then shaped and controlled by the system designer;
- Based on design techniques which encompass breakdown and repair, and can embrace incomplete specifications, uncertainly and error;
- Aware of what the default state is and able to return to it;
- Developed taking into account the value of variety, the selection of solutions from a pool of solutions and a willingness to repair.

Such systems will be more dynamic than the static systems of the past. Hence they will require dynamic design methods which focus on adaptability, change, communication and relationship. Current approaches which are constructivist direct developers towards the construction of ‘blueprints’ for system which fix processes and data. They become fossilized. While they might have impressions of soft parts like a fossilized cretaceous turtle, they are essentially dead and cannot adapt to changing environment, business or users.

4 DESIGN ISSUES

Creating the methods and IT artefacts to meet these grand challenges is essentially a design-science problem. It will involve the creation of new design processes which move away from the traditional, top-down architectural design approach. It will require the development of new constructs, models, methods and instantiations. Design-science is a problem solving paradigm with its roots in the sciences of the artificial. The purpose of applying design science will be to create IT artefacts which more closely meet the information requirements of organisations and individuals within the social and technical context. This will require new foundations and methods which enable the design of systems where the focus is on adaptability, malleability, extension and robustness rather than completeness and correctness. The research should address the design-science guidelines of Hevner et al (2004). New artefacts should be produced which demonstrate the characteristics expected of adaptable information systems. The inadequacy of many information systems to meet business needs is clear. However, more work will be needed to define the business problems which are most suited for the application of information systems based on different approaches to analysis. Evaluation will involve an examination of the utility of the design approaches based on an alternative systems design paradigm. The research contribution arises from challenges to current approaches to systems analysis and design. Research rigor will involve the assessment of the generalisability of the foundations and methodologies developed to a wide range of problems. Measurements will have to be developed which differ from existing development metrics. Measurements will be needed of adaptability, of extent and ability to repair, and of ease of extension of resulting systems. Hevner et al (2004) point out that design is a search process, defined as their guideline 6. In looking for solutions to grand challenges, a search will be needed. The second half of this paper will suggest that a prime search area for solutions to adaptable systems lies in the realm of biological phenomena.

In tackling this as a design-science problem, a variety of design issues will be faced, particularly in describing the design. Consideration will have to be given to:

- Defining components and defining dynamic strategies for their aggregation to provide adaptable solutions to business problems.
- Describing design strategies for emergent design where the final design is not certain and will arise from interactions between smaller units. In other words, how do we portray bottom-up design where the problem is incomplete and the solution contains a large amount of uncertainty and loose ends?
- Describing design approaches where the focus is on selection from a pool of components or designs.
- Effectively describing interfaces, interaction at the interface and the evolvability of the interface.
- Describing design in terms of extendable networks rather than fully scoped hierarchies.

These challenges and issues will require new theoretical foundations. It may ultimately apply new mathematical foundations from the field of complexity science, synchronocity, and graph theory. It may require new models to represent the design data. It will also require new design science methodologies for evaluating resulting artifacts.

The previous sections of this paper have outlined the challenges and resulting design issues. To seek design solution for these challenges requires a searching step involving the defining of a search space. The second half of this paper discusses a possible solution space. The following sections briefly review the range of application of biological inspiration to computing and then concentrates on the use of biological design at the molecular level to generate design-science ideas for systems development.

5 BIOLOGICAL INSPIRATION IN COMPUTING

It is noteworthy that most of the challenges discussed above have been solved by biological systems. Biological systems are eminently adaptable to environmental change. They combine strongly conserved design at the macro and micro level with flexibility, individuality and vigorous response to short term and long term environmental change. They combine simple building blocks with complex behaviour and structure. They conserve correct working design structure thorough extensive attention to repair and the generation of immense variety. They use strongly hierarchical control mechanisms which are centralized and at the same time use networks of control to create dynamic design where there is a significant amount of autonomy and individualism.

Biological systems differentiate well between the developmental phase when rapid changes occur within the embryo and juvenile and the maintenance phase where the fully developed organism interacts with its environment. In adaptation terms, biological systems address the dichotomy between stasis and change. Structure and function of species is strongly conserved, and yet evolutionary mechanisms permit rapid change if environmental stresses dictate such a need (Rutherford and Lindquist, 1998).Biological systems combine economy and reuse of specification with a level of redundancy and duplication that supports requisite variety. Genes coding for some proteins can be read in several ways such that sequences participate in several different proteins. Equally some sequences are duplicated, and there are multiple occurrences of some genes in certain organisms.

Biological mechanisms provide a wide inspirational based for moving systems and systems development out of an engineering and formal correctness paradigm into a more flexible adaptation and repair paradigm. As such a wide range of biological phenomena should be explored and the design concepts applied to systems development

The application of biological concepts to computer science is not new (Forbes, 2004, De Castro, L.N. & Von Zuben, F.J.,2004; George et al, 2002, IBM,2001). Genetic algorithms and genetic programming are derived from basic evolutionary principles of recombination, mutation and selection for fitness. Neural networks use the analogy of networks in the brain as basis for solving problems, particular in, for example, pattern recognition. The recent focus on the Grand Challenges in research in computer science has highlighted the importance of using biological concepts to advance computer science.

Artificial immunology applies the concepts of the mechanisms of immunity in mammals towards the problem of providing immunity to computers. Autonomic computing takes the prime principles of the autonomic nervous system and applies them to distributed computer systems to develop system that can self-regulate, adapt to changing demands, self-repair and maintain their own homeostasis. Cellular computing uses ideas concerning communication between individuals in a biological system to develop computer programs exhibiting emergent behaviour. Computational vision draws on the mechanism of vision in biological systems linking sensory processing, perception and action in robot-based systems.

These applications are limited in the extent of detail. They tend to use the high level principles loosely, rather than exploring the detailed design apparent in biological systems. They also focus on the technical and program level, looking for technically implementible solutions to low level technical problems rather than addressing high level abstraction and particularly design.

At the molecular level there is a wide range of design concepts and details which could be drawn on in order to develop new approaches to systems development and support the paradigm shift necessary if computer systems with the adaptive characteristics describe above are to be designed. Molecular biology offers a wide range of design mechanisms which might be applied at various levels to computing. There is a specification defined in the DNA from which selected units are drawn for expression as proteins mediated by transcribed RNAs. This is subject to structural complexity in which genes can be broken down into subunits – introns and exons; and control complexity in which a large range of non-expressed RNAs and proteins act in control networks to select particular expressions from the range possible. An immense range of design and control mechanisms are apparent at the molecular and cellular level which have yet to be explored as a basis for system development techniques and as organising principles for developing systems which meet some of the grand challenges.

The remainder of this paper provides an exploratory discussion of some of the molecular mechanism of genetics and cellular development and the system development concepts that might be derived from them. The approach adopted is broadly a metaphor-driven approach to design (McBride, 2006) in which the source and target domains are pre-selected. The focus here is then on the search of the domain of molecular biology for constructs that might transfer to systems development. The complete design-science project will require the evaluation of those ideas for utility and effectiveness in supporting the design of new IT artefacts. The ultimate aim of such research is to develop new foundations in terms of theories, frameworks, instruments, constructs, models and methods which will support the development of systems where the emphasis is on adaptability, plasticity, autonomy and communication.

The stage of research represented below is fairly early in the design cycle. Bearing in mind the problem domain, four distinct concepts in molecular biology are explained and interpreted in systems development terms. The interpretation is presented as a number of high level guideline statements which could then be developed into foundations and then supported by methods.

6 METAPHORS FOR SYSTEMS DESIGN FROM MOLECULAR BIOLOGY

6.1 Introns and extrons

The human genome contains between 25,000 and 35,000 genes (Wolfsberg et al, 2001), that is sequences which can be transcribed into mRNA and translated into proteins. This accounts for a fraction of the 3.3 x 109 base pairs of the genome. Some of the DNA consists of groups of short highly repetitive sequences that are not transcribed. Moderately repeated sequences are also peppered across the genome. These consist of related but not identical sequences. Protein coding sequences are found in the non-repetitive DNA. Functional types are grouped together, for example, the haemoglobins. Functional clusters may contain inactive genes, called pseudogenes, which may have been rendered inactive by base mutations, or may be copies of the active mRNA, without introns, inserted back into the genome (Alberts et al,1998).

Each gene, giving rise to an expressed protein, may not be contiguous. Rather the gene consists of regions which code for parts of proteins, the exons, and regions that do not, the introns. Transcription results in a long precursor mRNA. Intron sequences are excised from the precursor mRNA and exons are spiced together to give the final mRNA which is translated into protein. Exons are joined together in the same order as they appear in the DNA. While many genes code for one protein, multiple types of protein can be derived from one gene by splicing out one or more exons along with the introns. Indeed, it is estimated that 60% of human genes may have multiple splicing forms. The purpose of introns is not really known. It may be suggested that they allow opportunity for variation such that basic units of protein function can be linked together to provide greater versatility (Lewin, 2000).

A loose interpretation of spliced genes suggests some possible guidelines in systems development:

- Analysis should focus on the smallest individual unit of processing.
- The genetic unit of analysis should include information, processes and control.
- Analysis should start with individual processes carried out by people.
- Alternative processes may be developed by cutting out some steps (i.e. splicing exons by cutting out introns.)
- Describe at level of individual activities first
- Initial output of design is a set of information system ‘genes’ which describe the smallest unit of processing.
- Systems can then be developed by building up small units of data, process and control.

- An individual unit of analysis describes a very small function or change.
- Units of analysis have start signals and indicators of when they might be turned off.
- Small units of processing may have varying affinities with out units.

Considering genes and the nature of genes suggest that the level of analysis needs to start with the small details. Rather than immediately abstracting and generalizing in systems analysis, a biological model of systems analysis may start with a concrete example of a small process. These small processes define small changes of state. The transcription and translation of a gene in a cell must result in a change in state. Similarly the activation of a small process in a system results in a change in state of the system.

6.2 Non-Coding RNAs

Many RNA molecules transcribed from the DNA do not code for proteins. Small RNA molecules act as control switches, switching on and off networks of expression, controlling developmental timing, controlling the alternative splicing of introns. The small non-coding RNAs constitute 75% of transcripts. They have high sequence specificity and are often very well conserved over evolutionary distances. They may be active in repair, removing unwanted sections of DNA. Some introns may be themselves small RNAs which have processing power. Non-coding RNAs may comprise the control architecture which enables enkaryotic complexity, underpinning differentiation and development.

Non-coding RNAs allow greater variability than protein networks. Differentiation of a sufficient critical mass to give rise to speciation could arise from cumulative changes in the much flexible non-coding RNA reaching a cusp or edge of chaos at which a bifurcation occurs which may become fixed as a species if the environment is right. Phenotypic and individual characteristics may be seen as emergent phenomena, resulting from the added informational value derived from interactive, dynamic molecular networks (Mattick, 2001).

An important aspect of non-coding RNAs is how they enable the creation of networks of interaction and control which while exhibiting stability also allow individuality. These networks are both dynamic in that they consist of constantly changing molecular reactions in which messages are passed between elements of the network and each element acts to change the structure of other members of the network. It is also stable in that the phenotypic or physically expressed outcomes of the dynamic network are stable and change only a little within set constraints.

Considering how small RNAs enable the creation of sensitive networks of control leads us to consider a system as a network of small processes interacting together to form a system in which behaviour emerges from the sum of the individual behaviour of the system's genes.

Hence the analysis process involves understanding the characteristics and properties of the individual genes, or individual business interactions within the system. Such a network can be expanded or shrunk as required. Elements can be added to the network which then influence existing parts of the networks, and elements can be taken away. The overall expressed behaviour will then be determined by the multiple interactions between controls, data and processes.

A network approach to developing an information system will also encourage repair and system change at the design level. Repairs can be built-in through the use of specific network elements which can act like inhibitor RNAs which turn off certain network processing elements or to isolate some element of the network. Such a philosophy of network control can be implemented at the design stage of a system or be designed to dynamically 'kick in' at certain points after the system has been implemented and is productive. Key control points can be implemented which trigger or shut off the network or parts of the network in response to environmental influences

In terms of integration and development-in-the-large, Existing application networks can be joined together at critical points. New application networks can be built separately and then latched onto existing systems through defined network joining points.

Biological networks in the cell are responsible for both systems maintenance and development. Biological development is specified in the genome and operates through temporal sequences of expression of proteins, controlled by proteins and RNAs. There is no reason why system design should not specify the progression of design such that a different information system is developed or emerges for each environmental situation.

6.3 Homeotic Genes

Homeotic genes control the identity of body segments. In Drosophila, the fruit fly, mutations in homeotic genes may cause antenna to be converted to legs or extra wings to be produced. In humans, mutations of certain homeotic genes results in limb or kidney defects. Homeotic genes code for DNA binding proteins that affect the transcription of other genes. The primary homeotic genes are clustered in the fruit fly in an order that reflects the spatial ordering in the fly itself. Homeotic genes may operate in cascades, in which one homeotic gene promotes the expression of further genes and results in the specific differentiation of a cell into a cell type whose memory of type may be a result of the expression of specific homeotic genes. In drosophila, three homeotic genes control ten different types of segment. The control mechanisms do not merely depend on the presence or absence of homeotic gene products, but also on little understood, persistent differences in non-expressed control regions around the homeotic genes.

Thus homeotic genes act as control components, controlling the development of specific domains, whether spatial such as limbs or functional such as liver or kidney. Thus domains are constructed in a hierarchical manner. Some homeotic genes may affect sub-domains that group to form larger structures.

The point about homeotic genes is that large structures – legs, muscle, bone, kidneys – are controlled by very small numbers of genes. Mutating a single gene can create new legs on a Drosophila. Net works of development processing have very narrow gateways. Homeotic genes control reproducible domains.

Homeotic genes give access to set patterns of development within particular domains. These domains at a species level and a functional structure level are surprisingly stable. A gorilla remains a gorilla and does not radically get redesigned to another species. Organ domains remain intact. A liver is the same across mammalian animals. Molecular structures such as ribosomes, mitochondria, and membranes are conserved across vast ranges of life. Hence these designs are well scope, defined within a fixed domain and eminently reusable across a wide range of biological systems.

Taking this idea forward gives a significant clue as to how systems development and design should be progressing:

- There should be significant research effort to identify and scope domains, to define the structure and function of systems and define the homeotic mechanism that might control their use in systems.
- Such an activity could be described as open systems design, where the focus on code in open systems is moved to a focus on domain development and the building of homeotic systems which might apply across many systems. What is the systems development equivalent of the heart, muscle, Golgi apparatus and cycle AMP, for example.
- Systems development effort should go into exploiting patterns at a higher level. Primarily patterns are associated with programming, programming structures of which model, view and controller is

an example. This is a concern with implementation rather then the design of processes and the expressing of business problems.

- Systems development should focus on defining, scoping and giving boundaries to key processes within the organization. These may form processing domains whose network of processing is controlled by key homeotic elements.

Biologically, domains may concern oxygenating blood, developing immunity, focussing light, and digestion. The evolved solutions to the domains problems are substantially stable. Similarly, solutions to problems within business domains may be stable. An e-commerce transaction system does not evolve in the large; changes and alterations as a result of organisational change will generally occur in the detail, rather than involving a catastrophic change in the business function. Indeed, it may be these small changes that consume the most maintenance time overall, rather than big system changes. Systems developers should look at designing domain systems which encompass the ability to change details of design, but not the overall domain concept, in response to external environmental changes.

6.4 Receptors

In biological systems, communication between cells and systems is mediated by receptors. These receptors, located in cell membranes, bind reversibly to specific chemicals, for example hormones. The effect of this binding will differ according to the cell type. In some cases, the binding of the hormone to the receptor endows the receptor with increased affinity for DNA. New transcription and new translation is triggered. This may operate in a cascade fashion, where the receptor triggers the production of a protein which itself triggers production of several proteins similar to the manner in which a cascade of homeotic proteins are expressed.

Binding of a hormone or ligand to a receptor may have other effects. Primarily, these include altering the voltage on a membrane and thus changing permeability to different minerals and triggering the production of cyclic AMP. Cyclic AMP is considered a ubitiquous intracellular mediator in that the effect of many receptors within cells is triggered by cyclic AMP production. While there are many different kinds of receptors with specificities for different ligands, there are only a few standard mechanisms by which the receptor triggers intercellular change.

In systems development, changes to domains, perhaps triggered through homeotic elements should be designed into the system. Key links between networked systems may be designed as information receptors which pick up signals from the environment or latch on to key elements in other networks and cause linked processing. Hence system design work, which should shift its focus much more to the interfaces and periphery of the systems, where the action really is, should include the design of connection, hooks or receptors that offer links between systems. These links should be dynamic and adaptable, being activated by the presence of another system nearby, the receiving of an environmental signal or the action of a user whose intervention may result in system activity. This intervention will include the reconfiguration of the system in a manner which will be expected in service-oriented computing where triggers, requests or signals are received which define the service required at a particular time and place and result the delivery of service by parts of a control network.

7 CONCLUSIONS

This paper has explored some of the demands being made of systems developers. It recognises that the paradigms are shifting to the extent that old approaches to systems development are being found to be wanting. However, while soft systems approaches to analysis are not finding currency and the UML based systems developments are limited in their ability to meet the demands of businesses. The paper then recognises that the nature of these problems requires that systems developers look beyond the traditional disciplines for ideas.

Using molecular biology as a source domain, a number of ideas emerge. Biological systems are able to both respond dynamically to environmental demand and conserve structure and function. This is clearly what information systems need. The paper demonstrates that there are possibilities for pursuing new ideas by exploring biology as a source of design ideas, not only at the detailed implementation level, but also at a higher design and analysis level.

References

Alberts,B; Bray,D, Lewis,J, Raff, M, Roberts, K and Watson,J.D (1998) The Molecular Biology of the Cell. Garland.

De Castro, L.N. and Von Zuben, F.J. (2004) Recent Developments in Biologically Inspired Computing. Idea Publishing, Hershey.

Forbes,N. (2004) Imitation of Life: How Biology Is Inspiring Computing. MIT Press

George,S., Evans,D. and Davidson,L. (2002) A Biologically-Inspired Programming Model for Self-Healing Systems. WOSS'02 November 18-19, Charleston,USA.

Hevner, A.R, March, S.T., Park, J and Ram,S (2004) Design Science in Information System Research. MIS Quarterly, 28 (1) 75-105

IBM (2001) Autonomic Computing: IBM's perspective on the State of Information Technology. http://www.ibm.com/research/autonomic.

Lewin,B. (2000) Genes VII Oxford

Mattick, J.S. (2001) Non-coding RNAs: the architects of eukaryotic complexity. EMBO Reports, 2(11) 986-991. Available at: http://embo-reports.oupjournals.org/cgi/content/full/2/11/986

McBride, N. (2006) Towards a metaphor-driven information systems design process. First International Conference on Design Science Research in Information Systems and Technology. Claremont Graduate University, California, February 24-25 2006.

Rutherford, S.L. and Lindquist,S. (1998) Hsp90 as a capacitor for morphological evolution. Nature 396, 336-342.

Wolfberg, T.G.; McEntyre,J and Schuler,G.D. (2001) Guide to the Draft Human Genome. Nature 409, 824 - 826.