Qualitative Reasoning in Integrative Biology: Evolving a Network Experiment Modeling System for Collaborative Multilevel Systems Research

Robert B. Trelease
Dept. of Pathology and Lab. Med.
Geffen School of Medicine, UCLA
Los Angeles, California, 90095 USA
E-mail: trelease@ucla.edu

Jack Park
Thinkalong Software
Box 61257
Palo Alto, California, 94306 USA
E-mail: jackpark@thinkalong.com

Abstract

We address important design considerations in effecting a network server-based experiment modeling system intended for integrative biology and biomedical research. We proceed to that end by following the progressive development of application features for our experimental qualitative process (QP) modeling system. Early work on a PC-based system for experimental biology focused on multiple-level visualizations of cellular immune responses, as well as receptor-mediated processes regulating protooncogene expression. Other practical applications focused on lymphocyte circulation and organopedesis, hyperbaric oxygenation effects on angiogenesis, macrophage migration, and fibroblast elaboration of ground substance. With the introduction of stabile Java technology in the mid-90s, work began on migrating the original PC-based modeling system environment (TSC) to a framework that would eventually support parallel execution of inference processes in a server-based environment. As the project has evolved, we have had to consider incorporating output functions of cooperative applications, issues for database and ontology importing, and technical challenges in representing level-unifying knowledge and experimental process rules.

1 Introduction

In integrative biology and medical research, new attention [2] has been focused on the need for integrating data, information and knowledge from multiple frames of reference and disciplines, in order to provide more comprehensive scientific models of how dynamical processes interact at molecular, cellular, organismal, behavioral, and population levels of representation. In genomics, for example, recent emphasis has been put on the need to accumulate usable ontologies of specific gene functions, in order to provide a foundation for understanding the dynamics of various biological processes[3].

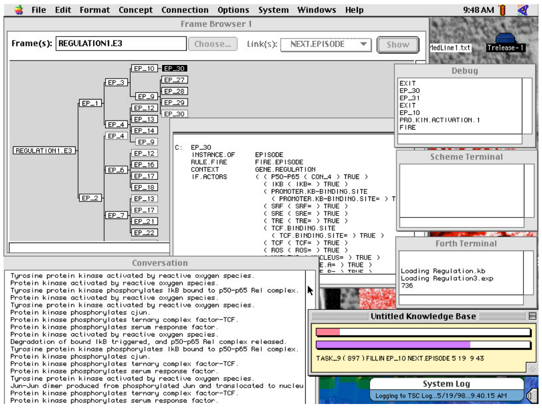
Practically applying quantitative methods to implementing process models at a few levels has lead to the development of useful 'in silico biology' programs such as the Virtual Cell, In Silico Cell, and Physiome. However, in this context, it has been openly stated that so much data, information and knowledge are being collected, that they far outstrip the ability of scientists and programmers to make practical integrative use of them. As one genomics pioneer noted recently [12], strictly quantitative approaches to in-silico biology immediately face an intractable combinatorial explosion of multiple levels of parallel equations exceeding the capabilities of current computer resources. More significantly, many biological process descriptions and relationships are largely qualitative and are quantitatively undefined or deterministically undefinable. Under such circumstances, it is worth considering qualitative reasoning (QR) for an alternative approaches for managing complexity in multilevel biological modeling.

During the early 1990s, we began the development of a qualitative process modeling system for running simulated experiments on biological processes, cells, and organisms[15][13]. As the early years passed, practical use of the World Wide Web burgeoned, and great new resources became available for 'doing science better' via Internet connectivity. We have thus seen that our qualitative modeling environment could evolve into a more powerful multidisciplinary collaborative environment for research ultimately supported by today's distributed computing networks, online databases, ontologies, and powerful, openly available information tools.

2 Methods and Design Issues: Early efforts with workstation-based qualitative modeling

Initial work focused on creating a broad-based hierarchical knowledge base system incorporating features of biological taxonomy as well as details of eukaryotic cell biology. In seeking to pioneer some applications of qualitative reasoning in biology, the senior author was particularly influenced by the concepts of QP theory (QPT, from Kenneth Forbus [4]) and qualitative simulation (QSIM, from Benjamin Kuipers [9][10]) as applied to physics problems. Lessons were also learned from Peter Karp's development of EcoCYC, a pioneering modeling-oriented ontology for microbial genetics [8].

For our biological modeling system, concepts, substances, cells, organisms, and multilevel processes were symbolically defined in an object-oriented environment known as TSC (The Scholar's Companion). The initial programming environment include underlying Pascal, Forth, and Scheme components, with knowledge bases coded in Scheme frames. Quantitative data were handled by numerical decoding process rules. This system was used successfully to model primary immune responses to bacterial and viral pathogens [16], as well as regulatory mechanisms for protooncogenes underlying specific hormonal, free-radical, and immune system signal transduction [14].



```
EXIT
EP_30
EP_31
EXIT
EP_10
PRO.KIN.ACTIVATION.1
FIRE
```

```
C:  EP_30
    INSTANCE.OF        EPISODE
    RULE.FIRE          FIRE.EPISODE
    CONTEXT            GENE.REGULATION
    IF.ACTORS          ( ( P50-P65 ( CON_4 ) TRUE )
                         ( IKB ( IKB= ) TRUE )
                         ( PROMOTER.KB-BINDING.SITE
                           ( PROMOTER.KB-BINDING.SITE= ) T
                         ( SRF ( SRF= ) TRUE )
                         ( SRE ( SRE= ) TRUE )
                         ( TRE ( TRE= ) TRUE )
                         ( TCF.BINDING.SITE
                           ( TCF.BINDING.SITE= ) TRUE )
                         ( TCF ( TCF= ) TRUE )
                         ( ROS ( ROS= ) TRUE )
                                    CLEUS= ) TRUE )
                                   E.A= ) TRUE )
```

```
Loading Regulation.kb
Loading Regulation3.exp
736
```

```
Tyrosine protein kinase activated by reactive oxygen species.
Protein kinase activated by reactive oxygen species.
Tyrosine protein kinase phosphorylates IkB bound to p50-p65 Rel complex.
Protein kinase activated by reactive oxygen species.
Tyrosine protein kinase activated by reactive oxygen species.
Protein kinase phosphorylates cjun.
Protein kinase phosphorylates ternary complex factor-TCF.
Protein kinase phosphorylates serum response factor.
Protein kinase activated by reactive oxygen species.
Degradation of bound IkB triggered, and p50-p65 Rel complex released.
Tyrosine protein kinase phosphorylates IkB bound to p50-p65 Rel complex.
Protein kinase phosphorylates cjun.
Protein kinase phosphorylates ternary complex factor-TCF.
Protein kinase phosphorylates serum response factor.
Tyrosine protein kinase activated by reactive oxygen species.
Jun-Jun dimer produced from phosphorylated Jun and translocated to nucleu
Protein kinase phosphorylates ternary complex factor-TCF.
Protein kinase phosphorylates serum response factor.
```

```
TASK_9 ( 897 ) FILLIN EP_10 NEXT.EPISODE 5 19  9 43
```

```
Logging to TSC Log...5/19/98...9.40.15 AM
```

Figure 1: TSC running a gene regulation experiment.

Other (unpublished) process models created with the initial TSC system simulated diapedesis and systemic circulation of lymphocytes (with organ-specific binding mediated by cell adhesion molecules), as well as macrophage migration, angiogenesis and wound healing under conditions of hyperbaric oxygenation. Beyond the research environment, the modeling system was also employed in trials in secondary school instruction for simulating scientific experiments and representing theories as diverse as those of immunity, ecology, and extinction. In one original instance, operating outside of the biological domain with a different knowledge base, TSC provided conjectures which supported the discovery of new process control rules in a polymer curing environment [1].

With the introduction of stabile, fast-executing Java versions in the mid-90s, the senior author began experimenting with a simple applet version of a QP modeling system. Using a transliteration of earlier TSC knowledge base content into the CLIPS-like Scheme dialect of JESS (Ernest Friedman-Hill's Java Expert System shell), basic published proto-oncogene experiments were repeated. In this simplified example of a Web-based simulation resource, only textual envisionments were generated, but they indicated the feasibility of building a multiplatform QP modeling system with modifications of a forward-chaining inference environment.

3 The evolution of the TSC QR environment

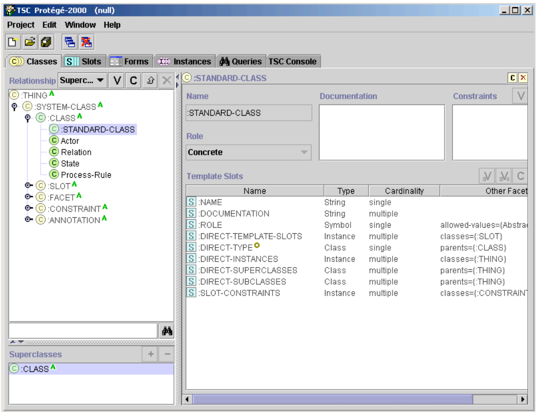
At the same time, TSC developer Jack Park began the process of translating TSC resources into Java. Java libraries for manipulation of list structures, coupled with built in automatic garbage collection of the Java working memory, made the translation relatively straight-forward. TSC used a supervisory agenda-based inference engine (envisionment builder), one that posted tasks to an agenda for software agents to perform. The move to Java made it possible to integrate the agenda functionality of TSC into a tuplespace Web portal being developed to support collaborative research and learning projects.

Tuplespace is the name given a kind of public repository blackboard architecture created by David Gelernter [6] , wherein tuple data or parameters (corresponding to non-relational database records) are used for communication between multiple active programs distributed over different machines. Tuplespace is thus well suited for agent coordination, and it has been implemented in a variety of ways in Linda, JavaSpaces [5], IBM's TSpaces, and TupleSpace4J. TupleSpace4J is the implementation being used for TopicSpaces, a Collaboratory being developed by Jack Park for research and learning projects.

The move to Java and coupling to a tuplespace implementation of the agenda management meant that the original inference engine itself was no longer needed for organizing agent functions. In Java-based TSC (called TSC4J), as tasks come into the agenda, they are distributed in tuplespace to be handled by specific agents which are, at once, available and programmed for the chosen task. In this context, tuplespace can also be viewed as associative memory for the QR system.

In essence, TSC4J is now a collection (collective) of agents intended for building models, studying those models, studying data flows in relation to models, noticing expectation failures (where data and model predictions fail to agree), and forming conjectures regarding the nature of expectation failures. Consistent with the original purpose of TSC and continuing its legacy in process discovery, TSC4J continues to serve a role as a modeling and analytical assistant.

TSC4J operates on its own internal information and knowledge structures, but includes import agents capable of translating and loading knowledge in many known ontology structures, e.g., RDF (Resource Description Framework), XTM (XML Topic Maps [11]), and OKBC (Open Knowledge Base Connectivity). At the same time, available export agents allow the results of TSC4J sessions to be written to any of those ontology representation frameworks. TSC4J is also being functionally integrated ("tabbed") with Stanford's Protégé 2000, allowing its use as a knowledge engineering tool for preparing new model ontologies.



:THING
:SYSTEM-CLASS
:CLASS
:STANDARD-CLASS
Actor
Relation
State
Process-Rule
:SLOT
:FACET
:CONSTRAINT
:ANNOTATION

Name: :STANDARD-CLASS

Role: Concrete

Documentation

Constraints

Template Slots

Name	Type	Cardinality	Other Facet
:NAME	String	single	
:DOCUMENTATION	String	multiple	
:ROLE	Symbol	single	allowed-values={Abstra
:DIRECT-TEMPLATE-SLOTS	Instance	multiple	classes={:SLOT}
:DIRECT-TYPE	Class	single	parents={:CLASS}
:DIRECT-INSTANCES	Instance	multiple	classes={:THING}
:DIRECT-SUPERCLASSES	Class	multiple	parents={:THING}
:DIRECT-SUBCLASSES	Class	multiple	parents={:THING}
:SLOT-CONSTRAINTS	Instance	multiple	classes={:CONSTRAINT

Figure 2: TSC running as a "TAB" with Protégé.

The proposed TSC4J Biology Network Experiment Modeling System (BioNEMS) would support not only collaborative modeling in multi-disciplinary professional research groups, but networked collaboration in science education and public enthusiast sectors. TSC4J development efforts are being managed within the larger TopicSpaces/TopicMaps Web resources.

4 Knowledge engineering issues in collaboration space

In seeking to incorporate existing online databases and ontologies, TSC4J BioNEMS faces several related challenges. At the outset, there is the functionality needed to match export and import agents to the task of translating knowledge contained in any of the currently available RDF-based (e.g., DAML/OIL, OWL) or OKBC-based (e.g., Ontolingua) formats, as well as to others which may be composed of tab- or comma-delimited text. Some of the necessary translation agents already exist within the TSC4J and TopicSpaces projects. The TopicSpaces Collaboratory engine combines XML topic maps for navigation with tuplespace for agent coordination, providing a powerful schema-neutral entity-attribute representation scheme. Tuples are capable of representing any object which can be decomposed into data fields or elements. Each input field gets a name, a value, and a value type, to which the tuple appends metadata on authorship, dates, security/privacy codes and identity of approved viewers in the case of private fields. The value in this simplified, canonical representation is that it can be easily mapped to/from other representation or serialization schemes, particularly tagged-language (XML, SGML, RDF, etc) serializations.

Federating BioNEMS research collaboratories by means of a unified XML topic map environment provides three important enhancements to traditional knowledge engineering tasks: 1) Rapid and precise navigation of the joint information resources created; 2) collaborative filtering by way of annotations and extensive linking and cross linking; 3) the ability to extend the topic map (knowledge base) by way of additional knowledge engineering processes and by way of linking to related materials found by mining the knowledge engineering work product.

Consider, for example, a BioNEMS model of a particular immune function. The subject of discussion is that immune function, and the model will contain representations of many episodes or sets of experimental conditions and state transitions for processes occurring during the course of a simulated experiment. Each episode is linked into the topic map.

The topic map provides immediate indexing of all actors, relations, states, and processes involved in the model, and those entities are all related, through typed topic map associations, the types of which are also topics which are indexed and available for discussion. When the system is used to its fullest capabilities by researchers, all references (such as journal articles, dissertation references, and other related technical information) will be integrated into the topic map. Reference information is thus linked directly to each and every simulation element for which there is an association.

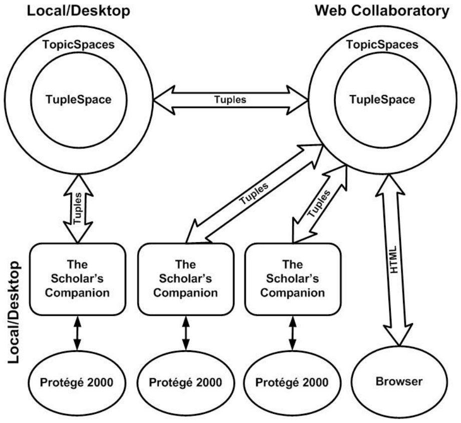


Figure 3: Conceptual block diagram of TSC4J components and BioNEMS.

TSC4J (with BioNEMS) is thus integrated into a Web-based collaboration space where all objects in models being built are also objects available for collaborative filtering, for discussion, and as content for learning experiences. Tuplespace provides the mechanism coordinating all agents, including those which provide TSC4J's modeling functionality. Thus, the topic map is itself continuously maintained by agents not directly related to TSC4J, but which are privy to

TSC4J's results as they appear in tuplespace. With this level of coordination of agents, it becomes possible, for example, for users to subscribe to different kinds of results and receive email reports of modeling progress.

5 Infrastructure issues for BioNEMS

Given the cross-platform Java servlet, XML, SQL and Web environments hosting BioNEMS, two initial configurations are anticipated. A simple local datamining approach would have BioNEMS installed on a small laboratory computing cluster, with hosting of imported, enhanced, mission-specific databases and ontologies locally as private resources. The larger extra-institutional collaboratory environment would be a publicly accessible resource supporting wide-scale scientific discovery efforts with publicly accessible ontologies and databases.

Although BioNEMS and its associated environment are essentially platform-nonspecific, we recognize that hardware design is an import issue for scientific collaboratory infrastructure intended to support intensive biological modeling. As part of this effort, we are investigating the performance of BioNEMS in a small, localized Unix-based supercomputing cluster configuration (gigabit interconnected multiprocessor servers).

6 Future directions

TSC4J and BioNEMS are very ambitious projects still in the midst of development. Given success of ongoing grant applications and more programming effort, we hope to be testing a functional network modeling system within the next year. Further development of the TSC4J BioNEMS system is expected to involve the continuous evolution of the knowledge engineering functionality. New agents will be created which treat ontologies much like collections of database entries and mine those entries for purposes of modeling and inference. Large and heterogeneous ontologies represent significant challenges to our ability to federate them in service of diverse bio-informatics communities of practice. Some ontologies will not easily submit to automated agent-based federation, and the need for humans in the knowledge engineering loop clearly will drive future enhancements in the system. The inclusion of Web-based communication via Topic Space enables multiple human experts to work synchronously or asynchronouslly on the same model or simulation application (human cluster processing).

At the same time in the infrastructure, Moore's Law should remain in effect: We see opportunities for continuous evolution of the code to take advantage of improved process threading on the hardward platforms hosting BioNEMS, of improved database systems, and of other advances in software technology. Since BioNEMS is an open source project which will include an SDK (systems development kit) for creating new agents, it will be possible for users in various bio-informatics communities to add new functionality to BioNEMS as their capabilities and needs evolve. The future of TSC4J BioNEMS hopefully involves continued expansion of system functions and modules which facilitate its continuing evolution with and by connected systems user communities.

References

[1] F.L. Abrams, *Process Discovery: Automated Process Development for the Control of Polymer Curing*, Doctoral Dissertation, University of Dayton, 1995.

[2] R. Altman and S. Raychauduri, Whole genome expression analysis: challenges beyond clustering, *Current Opinions on Structural Biology*, Vol. 11, 2001, pp. 340-347.

[3] M. Ashburner, and S. Lewis, On ontologies for biologists: the Gene Ontology–untangling the Web, In *'In Silico' Simulation of Biological Processes, Novartis Foundation Symposium 247*, pp. 66-83, John Wiley and Sons, New York, 2002.

[4] K.D. Forbus, Qualitative process theory, *Artificial Intelligence*, Vol. 24, 1984 pp. 85-168.

[5] E. Freeman, S. Hupfer, and K. Arnold, *JavaSpaces Principles, Patterns, and Practices*, Addison-Wesley, New York, 1999.

[6] D. Gelernter, *Mirror Worlds: Or the Day Software Puts the Universe in a Shoebox - How It Will Happen and What It Will Mean*, Oxford University Press, New York, 1992.

[7] P.D. Karp, Hypothesis formation as design, In J. Shrager and P. Langley, Eds., *Computational Models of Scientific Discovery and Theory Formation*, pp. 276-317, Morgan Kaufman, San Mateo, 1990.

[8] P.D. Karp, Frame representation and relational databases: Alternative information-management technologies for systematic biology, In R. Fortuner, Ed., *Advances in Computer Methods for Systematic Biology*, pp. 275-285, The Johns Hopkins University Press, Baltimore, 1993.

[9] B. Kuipers, Qualitative Simulation, *Artificial Intelligence*, Vol. 29, 1986, pp. 289-388.

[10] B. Kuipers, *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*, The MIT Press, Cambridge, 1994.

[11] J. Park and S. Hunting, *XML Topic Maps: Creating and Using Topic Maps for the Web*, Addison-Wesley, New York, 2002.

[12] B. Stewart, *An Interview with Jim Kent.* O'Reilly Bioinformatics Web site, http://www.oreillynet.com/pub/a/network/2002/12/10/kent.html, 2002.

[13] R.B. Trelease, Development of a feature-based knowledge base for biologically-based materials and processes. *USAF Contributive Research and Development*, Vol. 145, 1994, pp. 1-10.

[14] R.B. Trelease, R.A. Henderson, and J. Park, Experiment-based qualitative process modeling system for regulation of NF-kB and AP-1 protooncogenes, *Artificial Intelligence in Medicine*, Vol. 17, 1999, pp. 303-321.

[15] R.B. Trelease, and J. Park, Use of a heuristic discovery system in computer-based qualitative process modeling of biological systems, *Anatomical Record Supplement*, Vol. 1, 1993, pp. 115.

[16] R.B. Trelease, and J. Park, Qualitative process modeling of cell-cell-pathogen interactions in the immune system, *Computer Methods and Programs in Biomedicine*, Vol. 51, 1996, pp. 171-181.