



[landscape]

[projects]

[research]

[venues]

[prospectus]

[opportunities]

[personnel]

research

Theory as Product

Toward Autopoietic Database

Brett Stalbaum

Abstract

This paper examines in detail and positions two possible approaches to autopoietic database. The first is the cellular automata approach well known in the discipline of artificial life. The second is based on third order structural coupling drawn from autopoietic systems theory. The first approach is being born out in the present genealogical trajectory of database theory, design and development. Database is moving toward an object-oriented foundation, where objects express a high degree of autonomy. This trajectory still places heavy emphasis on traditional computer science notions of information processing and database as a discrete, organized collection of modeled data, and there are reasons to suspect that such databases are not yet fully autopoietic, although this is not ruled out. The second approach allows for analysis of social interactions of data that occur in poorly defined problem domains of active semiosis, and the mingling of variously encoded data and communication. Although in many ways antithetical to traditional notions of information processing and database ontology, the second (non-model) approach could be useful in evaluating systems where traditional data mining techniques (model based prediction) are impractical. Possible applications for this approach are speculated, and a number of art projects by C5 are cited as examples.

Introduction

Speculation about how an autopoietic database might be articulated involves parsing the same issues inherent in autopoietic theory generally. The processes utilized by systems that differentiate themselves from other systems on a continual basis through operational closure, (Maturana and Varela 89), and that produce and replace their own components in the process of interaction with their environment (structural coupling, *ibid.* 75), take place through a membrane containing the organization of the unity in question, thus allowing distinction between it and its environment. (*ibid.* 46) Such systems also exhibit great versatility and plasticity allowing expansion of possible behaviors, (*ibid.* 138) reproduction with conservation of adaptation, and motility. A basic question for any analysis (or design), based on autopoietic theory involves distinguishing the membrane, or the interface, where operational closure (inside) and structural coupling with an environment (outside) are expressed, because the membrane is the plane of distinction that allows any observation of plasticity, reproduction, and motility. Until we know where the membrane is, we can not analyze an autopoietic system, leave alone design one. This essay explores this issue in the context of database, speculating two paths toward autopoietic database in the context of the trajectory of contemporary database design.

Two approaches

There are at least two approaches that we might be tempted to follow toward autopoietic database. The first is the cellular automata approach common in the study of artificial life. The complexity we would expect such a database to exhibit might emerge from a multitude of very simple interactions between autopoietic data objects capable of retaining data and responding to queries. This approach focuses on the intentional simulation of autopoietic systems, and in fact the autopoietic nature of such collaborative knowledge systems has been advanced and even prototyped in the context of collaborative knowledge systems. (Cardon and Lesage) But a distinction should be made between the implementation of computational autopoiesis as proof of concept ('computational autopoiesis is possible'), and the potential practical applications of such a system, particularly as database. Varela's early work makes a strong case for the former, (McMullin

and Varela), but it is a substantially different problem to design autopoietic automata for implementation in database applications.

Varela's demonstration that autopoiesis can be computationally modeled is based on a minimal implementation of an artificial chemistry model. This is adequate to demonstrate the veracity of computational autopoiesis, but it does not necessarily indicate that autopoiesis can be effectively implemented computationally to perform work. (Nor was this a purpose of the experiment.) The internal goal, or intention, of an autopoietic system is restricted to the ongoing maintenance of its own organization, and this presents an obvious problem for anyone who wants to use autopoiesis for computation. Computation might indeed be a result of ongoing structural coupling between a collection of autopoietic unities and their environment, but such situations are not necessarily congruent with the solution to a specific problem or class of problems. In other words, we can not count on the natural drift inherent in living or life-like systems to most directly solve problems that are not the problems of conservation of adaptation and organization for those systems. Such systems have their own problems. The challenge in finding or engineering congruency between autopoietic systems and problems to yield solutions is enormous, rather like assisting the evolution salmonella in hopes of solving the traveling salesman problem. It is perhaps not impossible, but quite possibly not practical.

The second approach focuses on what Maturana and Varela call third order structural coupling. (Maturana and Varela 183) Here the focus is on the application of autopoietic theory as a conceptual description of living systems beyond the borders of traditional biology, theoretically extending autopoietic theory into the analysis of social systems. (First and second order structural coupling refer to single cell and meta-cellular systems respectively.) This includes more than cooperative couplings (social insects), or mating rituals, allowing for analysis of social interactions including linguistic, semiotic, and other coded symbolic domains such as computer networks. (Wittig)

The challenge involved in the analysis of existing social systems is quite different. Rather than finding or engineering congruency between the autopoietic systems and our problems, as in a collaborative knowledge system, what we are faced with is the difficulty of observation of a system that we are already implicated in, and whose discrete design can not be controlled via computer engineering practices. In the first approach, using autopoietic automata of our own design and specified in a manageable domain, we know where the membranes allowing distinction of identity are located, such that the system and its relations can be observed and knowledge processed. But constantly emergent third order structural coupling and the consensual domains they enable, provide no relatively easily observed membranes such as cell walls, skin, memory location, nor an interface to a Smalltalk(TM) object. Nor is there any clear delineation of input/output that may traverse third order membranes. How do we discover membranes that emerge in an ongoing manner through the use of language, such as takes place in a conversation or a network?

The existence of third order membranes depends upon the compositibility of relations between semiotic materials as they undergo the reciprocal process of ongoing structural coupling (mingling) in a languaging domain. (Slayton and Wittig) In domains such as language or computer networks, the challenge is to identify composable relations via as of yet undiscovered techniques that allow us to notice that a pattern of ongoing relations of semiosis are occurring between entities (unities) that are required for the conservation of autopoiesis in those entities. Once these relations are identified, not only would we know where the membranes are located, but with enough location data we would be able to develop a dynamic picture of the system that assists us in perceiving the system's model. Normally, data mining visualizations assume a model, but in visualizing third order structural coupling it is insight into an emergent or invisible

model that we seek. If workable, the related data mining approaches would provocatively diverge from the classic definition of information processing, where programs process raw data (input) into information (output), and move toward an approach that manifests visually an emergent manifestation of ongoing autopoiesis in language and other processed symbol systems. Herein might lay the conceptual basis for a practical non-model approach to knowledge discovery.

There is a second rather stunning conclusion that might be drawn from the above discussion. Not only does the second approach draw us away from the traditional definition of data processing, but at its extreme, it also succeeds the very notion of a database. The traditional notion of a database is an organized information store which can be queried and updated via a software application in a very controlled manner. But at an extreme, a database utilizing an autopoietic conceptual scheme would not have a data store at all. It would query the environment/knowledge domain directly. There would be no map, nor representation of the domain beyond the domain itself. The environment surrounding the application would itself be the database, and the autopoietic application would draw from that source via its structural coupling with that environment. Queries to the application would return concerns and reflections of the ontology of the environment, rather than internally consistent conclusions (facts) generated in the domain. Such applications could be useful in evaluating systems where traditional predictive modeling data mining techniques are initially impractical.

There is minimal need to weigh the two approaches against one another as they respectively constitute well developed areas of research. The first is in the traditional domain of artificial life research, which is a well developed field with much activity in the area of autopoietic systems. The second approach would be roughly in the domain of data mining visualization, and it is within this discipline that the second approach must prove itself. It is interesting to note that both alife and data mining place heavy emphasis on visualization techniques tending to emphasize semantic analysis in defined problem domains, rather than consideration of underlying organization and ontology of data in unstable domains where questions are poorly formed. In order to position the two related autopoietic approaches in the context of database, my technique will be to backtrack through the genealogy of database, in hopes that we can identify a pattern in the history of database development that may assist us in routing out a better speculative map toward autopoietic database.

A brief genealogy of database

A contemporary genealogy of database models can be viewed as the emergence of strategies for containing the parameters of the environment in which data is expected to function, in a way that facilitates the use of the data for an intended purpose. The most popularly familiar database model is based on the familiar file tree system, in which containment is expressed in parent/child hierarchies. File systems are designed to provide an interface to storage devices that provide ad hoc, flexible storage of variously encoded data files, programs to manipulate those files, and the operating system and its utilities. File systems allow new files, file containers, and navigational paths through the latter to be implemented with ease.

A direct result of the flexibility of storage for which file systems were designed is the difficulty involved in integrating data stored in different types of files utilized by different programs. This is described as the problem of structural dependence and data dependence. (Rob and Coronel 13) Access to a file is dependent on the file's organization, thus if the data's organization changes, the programs that use the data must change as well. It is a brittle system. The solution to this problem was the progressive separation of the physical implementation or file structure from the logical representation of the data. Once logical description and physical implementation are separated, (an interface built between them), changes to the

physical implementation of data can be implemented, eliminating the need to alter any applications utilizing the data. The genealogy of database pivots around this matter.

The strategies for separating the physical implementation from the logical representation are provided through a software interface of a Database Management System (DBMS). A DBMS is software that manages the storage and retrieval of data between application logic and the binary form in which it is stored. This allows the logical form of the data to be abstracted on one side of the DBMS, and physical storage details to be managed on the other. Logical abstraction is the goal in containment of implementation details, and different DBMS models express different containment strategies based on their history. The four major historical database models that emerged from this endeavor are Hierarchical, Network, Relational/Entity Relationship Model (ERM), and Object Oriented. Their containment strategies are described as follows.

A Hierarchical Database uses domain and name space containment. Hierarchical DBMS are a direct descendent of the original file systems, but the file structure and data types are strictly prescribed in comparison to the typical general purpose operating system directory, thus allowing modeling of hierarchical production systems. The hierarchical database was developed by North American Rockwell Corporation (later in collaboration with IBM), for the purpose of tracking the large number of parts for the Apollo mission. The model allows for records to be contained within a logical tree structure consisting of single parent nodes. One to one (1:1) and one to many (1:M) relationships are easy to represent, and this was highly congruent with hierarchical engineering structures common to aerospace manufacturing. (Rob and Coronel, 23) One of the disadvantages of the hierarchical model is that it is still somewhat brittle. Changes to the data change the navigational character of the database. The problem of structural dependence is improved, but not eliminated. Just as it is with the World Wide Web and its 404 messages, when information is deleted or moved, agencies expecting that information are left wanting.

The Network Database is a close descendent of the hierarchical database, with the difference that it allows more complex name space manipulation through the ability of each node or file to be owned by multiple parents. The domain constraints are similar to that of a hierarchical system, but the naming (the paths) can be made more complex such that it allows the modeling of more complicated data relations. This ability was one of the main reasons for creating the Network model. Indeed, because each node can have one or more parents, Many to Many (M:N) relationships are much easier to implement. The model dates to 1971 when the Conference on Data Systems Languages, (CODASYL) began a standardization effort, creating the Database Task group (DBTG). (Rob and Coronel, 28) The network database sought to overcome many of the problems of inflexibility inherent in hierarchical model, but nevertheless suffered from the problem of structural dependence mentioned above.

The Relational Database, developed by IBM researcher E.F. Codd in the early 1970's, utilizes attribute containment instead of name space containment. The relational model was developed as a reaction to the limitations of the hierarchical database, and can easily model 1:1, 1:M and M:N relationships. Data is logically modeled in tables of rows and columns, where the names given to the columns represent individual attributes of the records that are stored in rows. Tables can be related to one another using unique key values, thus allowing redundant data to be mitigated. By naming the attributes of data, and abstracting the location of the data into named tables representing entities, the relational database allows for strictly prescribed semantics and data typing. Through the use of common query language interfaces such as SQL, there is a stronger abstraction between the logical representation of data and the structure in which it is stored. Importantly, the relational database allows ad hoc queries to be formed, whereas

the hierarchical and network database models had to be designed with the questions that would be asked in mind. Research into natural language systems for the purpose of querying such databases is ongoing.

In an Object-Oriented Database, containment is specified by a public interface to data objects that are instantiated based upon a hierarchical system of data classes in an inheritance relationship. The design of the class hierarchy implements the data model, and specifies a public interface to object functions that return attribute values for individual data objects. Objects have private internal relations (private functions and data) which are independent of their exterior (or public) interface, although they are triggered by calls to the public functions. OO database objects also live in an environment of other database objects, and have the ability to message between objects (in order to implement the query functions of the DBMS.) The object oriented methodology emerged from Xerox Parc research into object oriented programming in the 1970's (the SmallTalk language), from which concepts were applied to database design by M. Hammer, D. McLeod in 1981. (Hammer and McLeod) In the relational model, only the relationships between the entities are included. An object extends this to encapsulate information about the relationships between attributes, and relationships between other objects. These are the "Basic building blocks for autonomous structures." (Rob and Coronel, 40)

Obvious trends in the ongoing development of database include a move toward greater abstraction between the application using the data and the data's form, a more sophisticated interface allowing more complex, ad hoc, or even conversational queries, and in the object oriented database, an adjustment to accommodate increasingly complex data types through the implementation of autonomous objects in conversation with one another.

Speculative direction toward autopoietic database

A DBMS is a kind of membrane. It transforms interaction on the outside of the interface (a query) into internal messages that trigger a response. The genealogy of database models indicates a clear motion in the direction of autonomous agent systems. But does this general direction of the systems we are observing provide room for all of the concepts outlined in Maturana and Varela's scheme for autopoietic systems?

The first approach, artificial life systems, seems to be in the process of being naturally borne out in the object oriented approach to database. Data objects may indeed exhibit operational closure and something like structural coupling through a membrane/interface, but do they produce and replace their own components, exhibit significant plasticity, and reproduce? The answer, for now, is no. It is not clear that any but some very specialized simulations (McMullin and Varela), produce and replace their own components, and true reproduction is much more complicated than copying or replication (Maturana and Varela, 59-61), both of which are common in information technology. Regarding plasticity, all database systems today presuppose a data model, and because the model is specified to perform a specific function, it will display a degree of rigidity. You can not ask your sales database about God, unless God is a customer. An additional question is how do database objects express motility? Other than the obvious fact that facts move around the world traversing data networks, it is not clear that autopoietic systems exist in motion, in part because we currently have no way of seeing them as they may emerge in third order structural coupling. It seems likely that with advances in Artificial Intelligence, alife, and mobile object programming techniques, such questions could be answered affirmatively.

The second approach, by contrast, only assumes that autopoiesis takes place in systems of active semiosis and the mingling of variously encoded data, information, and communication. Here we do not seek to design autopoietic systems, but rather seek tools to observe them. While it is both embryonic and

potentially an anathema to traditional computer science, the use of non-model systems for the purposes of uncovering models, or querying uncertain or unexplored systems, may prove potentially useful in applications where uncertainty and instability in the solution domain obfuscate traditional approaches to problem solving. Such applications might apply to understanding and tracking semiotic or linguistic interactions between various entities that produce conversations, and exposing any potentially important but presently invisible meta-knowledge those materials might express over time. Just a few potential applications include Groupware, Teamware, and collaborative applications, GIS analysis of battlefield operations, dynamic strategy management in business and team sports (particularly eco-challenge), social economics and information mediated markets, the study of communication and interrelationships between humans and animal populations, relationships between ecosystems, weather prediction and control, and intelligence operations such as surveillance systems on networks.

Work done in and around C5 demonstrate such non-model approaches. 16 Sessions (Walker Art Center) is a data mingling project that entangles biometric data with the internet's IP space to expose information relations. C5's 1:1 project (managed by Lisa Jevbratt), "includes the creation, maintenance, and visualization of the C5 IP database, containing the IP addresses to all hosts on the world wide web." (C5, 1) Visualizations generated by the project allow users to navigate the web through alternative interfaces that tie to the ontology of the internet's IP addressing scheme. SoftSub, (managed by Steve Durie), analyzes the directory structures of individual users, seeking organizational styles and structures that reflect larger communities of organizations that transcend the intentions of individual users. (C5, 2) Lisa Jevbratt's "Mapping the Web Infome" is also a fine example of a system that addresses poorly defined problems in an uncertain and unstable problem domain: web searching. In her curatorial statement, Jevbratt describes a "group of people in a dark room fumbling around not knowing what is in the room, how the room looks or what they are looking for." (Jevbratt) This is the type of situation in which non-model autopoietic database systems might provide compelling and practical solutions to knowledge acquisition.

References

Cardon, Alain and Lesage, Franck *Toward Adaptive Information Systems: considering concern and intentionality Eleventh Workshop on Knowledge Acquisition, Modeling and Management*, Banff, Alberta, Canada, Saturday 18th to Thursday 23rd April, 1998.

<http://ksi.cpsc.ucalgary.ca/KAW/KAW98/cardon/>

C5 Corporation (1), 1:1, New Langton Arts: The Bay Area Award Show
August 11 - September 25, 1999.

<http://www.c5corp.com/projects/1to1/newlangton.shtml>

C5 Corporation (2), SoftSub Research Project, An organizational visualization and mapping project.

<http://www.c5corp.com/softsub/>

Hammer M., McLeod D. *Database Description with SDM: A Semantic Database Model* ACM TODS 6(3): 351-386 (1981).

<http://www.acm.org/pubs/articles/journals/tods/1981-6-3/p351-hammer/p351-hammer.pdf>

Jevbratt, Lisa, *Mapping the Web Infome*, a net art endeavor developed in conjunction with the exhibition LifeLike at New Langton Arts gallery in San Francisco, June 27- July 28, 2001.

<http://dma.sjsu.edu/jevbratt/lifelike/>
http://dma.sjsu.edu/jevbratt/lifelike/curatorial_statement.html

Maturana, Humberto R., and Varela, Francisco J., *The Tree of Knowledge - The Biological Roots of Human Understanding*, 1987 Shambhala Publications, Boston Massachusetts.

McMullin, Barry and Varela, Francisco J. *Rediscovering Computational Autopoiesis*, February 1997, Santa Fe Institute Working Paper 97-02-012.
<http://www.santafe.edu/sfi/publications/Working-Papers/97-02-012/>

Rob, Peter and Coronel, Carlos, *Database Systems - Design, Implementation, and Management*, 2000 Course Technology, Cambridge Massachusetts.

Slayton, Joel and Wittig, Geri *Ontology of Organization as System*, Switch - the new media journal of the CADRE digital media laboratory, Fall 1999, Vol 5 Num 3.
<http://switch.sjsu.edu/web/v5n3/F-1.html>

Walker Art Center *16 Sessions*, a C5 project, commissioned by the Walker Art Center for the *Shock of the View* online exhibition, 02.17.99 thru 03.09.99. Curated by Steve Deitz.
<http://www.walkerart.org/salons/shockoftheview/hybrid/hybrid1.html>

Wittig, Geri *Situated and Distributed Knowledge Production in Network Space*, Switch - the new media journal of the CADRE digital media laboratory, Fall 2000, Vol 6 Num 2.
<http://switch.sjsu.edu/v6n2/articles/wittig.html>

[\[Home \]](#) [\[Research \]](#)