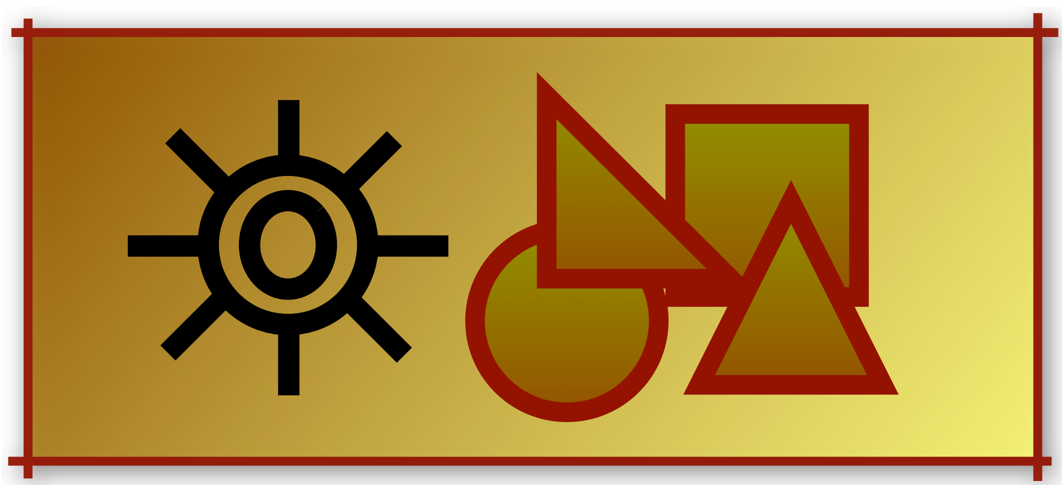
DESIGNING AND BUILDING THRIVING SYSTEMS

Project White Paper

February 18, 2009

PROMOTING LIFE USING THE RELATIONAL PARADIGM



Leslie J. Waguespack, Jr. Ph.D.

Professor of Computer Information Systems

Bentley University

Waltham, Massachusetts 02154-4705 • telephone: 781-891-2584 • 978-779-5322 • LWaguespack@Bentley.edu

Promoting Life

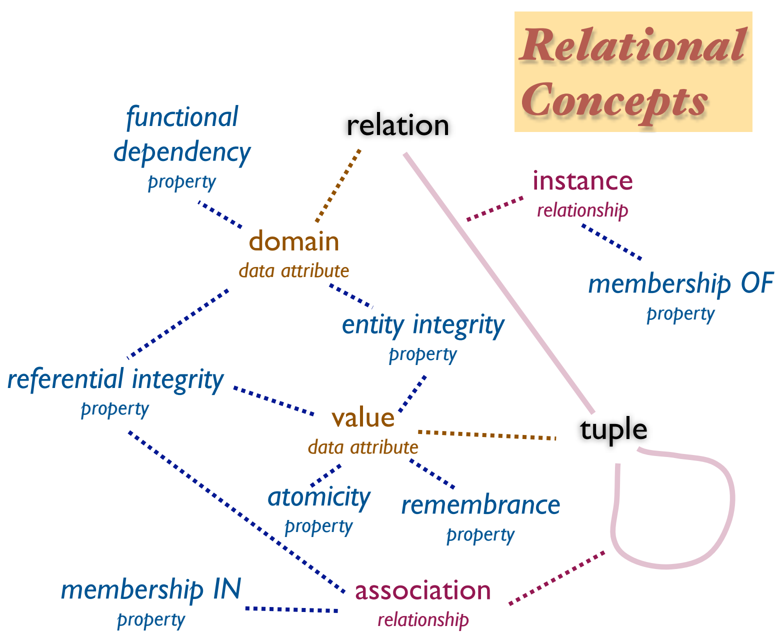
Using the Relational Paradigm

The concepts of system ecology and life preserving transformations espoused by Christopher Alexander can be integrated into any building process where choices are made. To further illustrate this integration this white paper describes how these concepts can be realized in and are facilitated by the relational paradigm and engineering processes associated with it.

The first step in this illustration is a survey of the defining characteristics of the relational paradigm independent of any of the programming languages used to represent or implement its features. The discussion begins with an ontology describing the paradigm's concepts and its vocabulary. That is followed up by a review of each of the life preserving choice properties and the opportunity that the relational paradigm provides for applying them.

An Ontology of the Relational Paradigm

This ontology is consistent with the practice in computer science and information science categorizing a domain of concepts (i.e. individuals, attributes, classes and relationships). As before in our examination of the object-oriented paradigm this ontology of the relational paradigm attempts to eschew the vestiges of implementation languages and development methodologies in order to expose the core nature and value of relational concepts based upon the work of E.F. Codd.



The Relational Ontology

The Relational ontology is arranged as follows:

Individuals
- Tuple

Attributes
- Data Attributes

Classes
- Relation

Relationships
- Behavioral Relationships
  - Functional Dependency
    - Entity Integrity
  - Association
    - Relational Operations
    - Join Compatibility
    - Referential Integrity
  - Normalization
    - First Normal Form
    - Second Normal Form
    - Third Normal Form

Individuals – The most concrete concept in the relational paradigm is the tuple.

Tuple – A tuple corresponds 1-1 with a single concept of reality that it represents. A tuple collects the facts that identify it as a single concept and the facts most closely identified with it.

Attributes – Attributes are those characteristics (facts) that describe a tuple. In the relational paradigm attributes define data characteristics - each of which has a static and dynamic form. A prescribed set of attributes defines what is called the structure of a tuple. From inception to extinction the structure of a tuple is immutable. The number of attributes in a tuple is called its degree.

Data Attributes – Data attributes store information (data) in the tuple and implement the property of remembrance. Remembrance is manifest in each attribute dynamically as "what is remembered," a particular data attribute value particular to each tuple derived from a data attribute domain that statically defines "what can be remembered," the possible values of the attribute.

Classes – The relational paradigm groups individuals into a collection called a relation. The relation corresponds directly with its mathematical antecedent where attribute values within each tuple reflect a correspondence with the coincidence of facts in the "real world," a correspondence (attribute relationship) that is shared by every tuple in that relation.

Relation – The relation concept combines both a definition of structure and the collection of tuple(s) based on that structure. A relation is defined as a fixed set of data attribute domains. Every tuple is an instance of a specific relation shares the same static structure defined by that relation with every other tuple of that relation. The relation concept thereby fuses the existence of the tuples to that of their relation; tuples cannot exist independent of their defining relation. Tuples are said to be members of their relation. Tuples are added to or deleted from their relation. The order of attributes in a relation is insignificant except that the order is consistent for all tuples. A relation is also commonly called a table and each of its instances, a row. The collection of data attribute value(s) for a particular data attribute from every row in a table is called a column.

Relationships – Relationships in the relational paradigm are based on the property of remembrance and the juxtaposition of data attribute values in one or more tuples in the same or across relations.

Behavioral Relationships – The behavioral relationships are all based upon the data attribute value(s) and which values are permitted to coexist in and across tuples and relations.

Functional Dependency – In a relation a data attribute is functionally dependent on another data attribute when for any particular value of the first there is always a single value for the second for any tuple. In other words, the value of the second data attribute is determined by the value of the first; the first attribute is sometimes called the determinant. Functional dependency expresses the informational integrity of relations.

Entity Integrity – Entity integrity defines the two-fold quality of tuple uniqueness in a relation: a) every tuple in a relation is distinct in some data attribute value(s) from every other tuple in that relation or symmetrically, b) there is a designated subset of data attributes (column(s)) called the primary key such that the data attribute value(s) of those data attribute(s) in that relation is distinct for all tuples and no values among them may be null (a value which is unknown and incomparable to any other value). There may be more than one subset of data attributes with the value characteristics of the primary key (each called a candidate key) but only one is designated as the primary key.

Association – An association is a relationship between tuples in the same or different relations. Tuples are intrinsically separable by way of entity integrity. At the same time, humans are compelled to categorize their experience of things in the physical world by superimposing groupings that collect tuples into sets. Tuples become members in a group based upon data attribute value(s). This property is called membership IN. This property also permits humans to identify a tuple that is not in a set (i.e. discrimination). (Membership IN an association is distinct from membership OF a relation which is intrinsic by way of instance relationship.)

Relational Operations – Membership IN is realized through relational operations keying on relation structure and values. Each relational operation produces a real or virtual relation as its result. The selection operation retrieves tuple(s) based upon a selection predicate testing data attribute value(s) to determine whether each tuple is or is not in the set. Selection predicates are based on any boolean comparison including constant values or values referenced in data attribute value(s). The projection operation copies all the data attribute value(s) for a particular column(s). Association between relations (or a relation and itself) is based upon relating (matching) data attribute values in tuples of one relation with those of another. The join operation pairs every combination of tuples from one relation with those of another relation and copies the data attribute values from the pairs where the pairing satisfies a selection predicate. This relational operation is called join because facts from two sources are joined in the result.

Join Compatibility – Join compatibility requires that the values involved in comparisons (i.e. selection predicates) whether constants or data attribute values derive from the same data attribute domain.

Referential Integrity – When relations are devised such that a tuple in one relation predisposes the existence of (owns) tuple(s) in another, the data attribute(s) of the second required to join the relations is called a foreign key. Referential integrity asserts that any value found in the data value attribute(s) of a foreign key must appear in a tuple of the first relation as the value of a candidate key or itself be null.

Normalization – Relational model consistency depends on the semantic concurrence of the behavioral relationships and the objectives of the database modeler, the intension, (rather than the accident of a relation's contents at any particular instant, its extension). The integrity properties defined above enable the database modeler to devise a structure and behavior of relations that avoid semantic discord called anomalies, the unintended loss or modification of information by relational operations. Relations designed to avoid certain kinds of anomalies are said to be normalized or in normal form. Normalization is the arrangement of data attributes and their relationships among relation structures to prevent particular anomalies.

First Normal Form – First Normal Form asserts that every data attribute value is atomic, indivisible in value or form and may not be operated upon except as a whole and single value.

Second Normal Form – Second Normal Form is first normal form and asserts that every data attribute value not in the primary key is fully functionally dependent upon the primary key. ("Fully" means applying to every data attribute of the primary key.)

Third Normal Form – Third Normal Form presupposes first and second normal forms and asserts that no data attribute outside the primary key is transitively dependent upon the primary key. ("Transitively" means an attribute(s) functionally dependent upon an attribute functionally dependent upon an attribute (. . .) functionally dependent on the primary key.)

Modeling Living Structure in the Relational Paradigm

Christopher Alexander's theories of wholeness and centers explain that the degree of wholeness derives from the strength or intensity that each center contributes to the whole. The strength or intensity of a center results from the intensity of the various fifteen properties that he attributes to the centers. Recall that centers were recast as choices to map Alexander's theories of physical architecture onto the domain of information systems, their models and architecture. The table below summarizes the fifteen properties and their interrelationships along with corresponding choice properties.

The discussion that follows examines each of the choice properties in turn and how each property relates to and may be facilitated by the relational paradigm. As these items derive from Alexander's property list which was parsed from an experience of physical architecture it is not surprising that the items in this list are often deeply interwoven in the promotion of life in the more abstract world of information systems.

	Alexander's Property	supporting properties															Choice Property
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	Levels of Scale		*	*			*			*							Stepwise Refinement
2	Strong Centers				*			*		*	*			*		*	Cohesion
3	Boundaries		*		*			*	*	*	*						Encapsulation
4	Alternating Repetition		*			*	*		*	*						*	Extensibility
5	Positive Space	*	*	*			*	*		*		*		*			Modularization
6	Good Shape	*	*			*	*		*		*		*		*		Correctness
7	Local Symmetries	*				*				*				*			Transparency
8	Deep Interlock and Ambiguity				*	*				*		*	*			*	Composition of Function
9	Contrast			*		*			*		*			*		*	Identity
10	Gradients	*	*					*		*		*	*			*	Scale
11	Roughness		*			*	*				*				*	*	User Friendliness
12	Echoes	*					*	*			*	*				*	Patterns
13	The Void	*		*		*		*		*					*		Programmability
14	Simplicity and Inner Calm						*	*					*	*		*	Reliability
15	Not Separateness			*		*			*		*	*		*	*		Elegance

Properties of Centers Mapped to Choice Properties

1. STEPWISE REFINEMENT reliant upon Cohesion, Encapsulation, Correctness and Identity *(derived from Alexander's Levels of Scale reliant upon Strong Centers, Boundaries, Good Shape and Contrast)* incorporated in the choice by the modeling action "to elaborate:"

In the relational paradigm the strength of the information that a relation represents derives from the choice of attributes and their interdependence that form that entity of knowledge. Each relation depicts a cohesive, encapsulated and distinct segment of knowledge. Each instance of that knowledge depends on its distinguishable identity: tuple by tuple. The scope of knowledge included in any particular model is constructed by the aggregation of these distinct segments interwoven through their explicit relationships. A whole model is built up stepwise as the "subset of the universe" chosen for the model (its intension) is systematically surveyed, cataloged and defined in the collection of relations. Each relation's integrity is achieved through its independent correctness separate and distinct except for those relations with which is maintains foreign key relationships. But the correctness of the whole proceeds from the assembly of the entire set of relations that together describe the reach of a model's responsibilities.

2. COHESION reliant upon Extensibility, Transparency, Identity, Scale, Programmability, Elegance *(derived from Alexander's Strong Centers reliant upon Alternating Repetition, Local Symmetries, Contrast, Gradients, The Void and Not Separateness)* incorporated in the choice by the modeling action "to factor:"

Each relation serves a separate role in the responsibility of representing domain knowledge. Relations reflect identity as they distinctly capture and represent concepts in the form of facts collected to represent cogent, clearly defined information. The tuples within relations similarly represent cogent, unambiguously defined instances of reality patterned after the attribute structure of their containing relation while by virtue of their entity integrity they remain distinct from any other tuple therein. The population of tuples in a relation over time reflect the ebb and flow of experience that the relation captures in the dynamics of the represented reality (the extension). The attribute structure of the relation as a template for each of its tuples ensures that the experience remains comparable and thus understandable regardless of the number of instances that experience produces. Functional dependency and its role in normalization assure that each relation represents an unambiguous and atomic division of knowledge in the modeling space. The result is a collection of distinct knowledge experiences bound together by a structure that both explains the significance of each instance and enables the analysis of that experience in terms of the whole reality that the relation captures.

3. ENCAPSULATION reliant upon Cohesion, Extensibility, Transparency, Composition of Function, Identity, and Scale *(derived from Alexander's Boundaries reliant upon Strong Centers, Alternating Repetition, Local Symmetries, Deep Interlock and Ambiguity, Contrast and Gradients)* incorporated in the choice by the modeling action "to encapsulate:"

In the relational paradigm the individual relation assumes the responsibility for capturing and defining the "reality," the "facts," the modeler chooses to instill in a model. The modeler's intension is represented in the structure of facts that each of its instances must be able to remember. Each instance of the relation remembers by way of the data attribute value set in each tuple. The truthfulness of individual tuples can thus be independently established. An important part of the reality captured in each tuple is its individuality and the uniqueness of the information that it remembers in its data attribute values, its entity integrity. This individuality is determined solely by the values encapsulated therein dependent on no other information or relationships; as characterized by Second Normal Form.

4. EXTENSIBILITY reliant upon Cohesion, Modularization, Correctness, Composition of Function, Identity and Elegance *(derived from Alexander's Alternating Repetition reliant upon Strong Centers, Positive Space, Good Shape, Deep Interlock and Ambiguity, Contrast and Not Separateness)* incorporated in the choice by the modeling action "to render extendable:"

Although each relation (down to the individual tuple) represents an independent depiction of reality in a relational model, more complex information is possible through the relationships that associate relations. Associations permit the depiction of more elaborate descriptions of a model's responsibilities. Associations depict correspondence, interdependence or even ownership of concepts between and among relations. These associations are employed through the relational operators that combine or collect facts resident in multiple relations and render them correlated, organized and/or extracted as a consistent but new representation of knowledge contained in the model.

5. MODULARIZATION reliant upon Stepwise Refinement, Cohesion, Encapsulation, Correctness, Transparency, Identity, User Friendliness and Programmability ***(derived from Alexander's Positive Space reliant upon Levels of Scale, Strong Centers, Boundaries, Good Shape, Local Symmetries, Contrast, Roughness and The Void)*** **incorporated in the choice by the modeling action "to modularize:"**

By the nature of depicting model knowledge in a collection of individual relations that knowledge is subdivided and compartmentalized. Furthermore the process of normalization assures that the intension depicted by individual relations and combinations of relations through their associations is neither ambiguous, redundant nor inconsistent. The compartmentalization of knowledge not only affords stakeholders a clearer view of relations individually, but exposes the opportunities to safely recombine that knowledge through relational operations. This cohesion that distinguishes each relation's role in the intension of the model also segregates the concerns that accomplish the model's responsibilities and permit attention to be focused on relevant subsets of the overall model's complexity.

6. CORRECTNESS reliant upon Stepwise Refinement, Cohesion, Modularization, Correctness, Composition of Function, Scale, Patterns and Reliability ***(derived from Alexander's Good Shape reliant upon Levels of Scale, Strong Centers, Positive Space, Good Shape, Deep Interlock and Ambiguity, Gradients, Echoes and Simplicity and Inner Calm)*** **incorporated in the choice by the modeling action "to align:"**

Entity integrity, referential integrity and normalization directly support a relational model's fidelity to the modeler's intension. Entity integrity assures that the uniqueness of each depiction of reality (extension) is enforced by the structure of the relation, intension, (the attribute set, their respective data attribute domains and the respective functional dependencies). The specification of that subset of attributes that will always contain a unique (combination of) value(s) defines the discriminating characteristics of that knowledge (the primary key) – the conformance to which is easily tested and thus protected. Referential integrity assures not only that data attribute values conform to the intension of their relation's data attribute domain but further, to the modeled intension of associations between tuples including the ownership relationship between relations. Normalization extends the assurance of fidelity (model to the modeler's intension) by assuring that the interrelationship among data attribute values not only supports entity integrity and referential integrity but, also inhibits the accidental loss of model knowledge (anomalies) through the action of relational operators.

7. TRANSPARENCY reliant upon Stepwise Refinement, Modularization, Identity and Programmability ***(derived from Alexander's Local Symmetries reliant upon Levels of Scale, Strong Centers, Positive Space, Contrast and The Void)*** **incorporated in the choice by the modeling action "to expose:"**

The relational paradigm facilitates transparency in two obvious respects. Inspecting the relevant data attribute values is sufficient to assess every aspect of integrity whether entity integrity or referential integrity. These same continuously accessible values form the basis of all relationships among data attribute values or among relations. The consistency of each and every data attribute value can be certified. At any time before or after any and every relational database operation we can verify concurrence with the time independent definition of intension given by the data attribute set and their respective data attribute domains along with the designation of candidate and foreign keys. There are no implied or hidden definitions of association or dependence. Every aspect of tuple or relation fidelity is discerned through self-evident information. The result of any relational operator is determined solely by the data attribute values of the relations involved.

8. COMPOSITION of Function reliant upon Extensibility, Modularization, Identity and Programmability *(derived from Alexander's Deep Interlock and Ambiguity reliant upon Alternating Repetition, Positive Space, Contrast and The Void)* incorporated in the choice by the modeling action "to assemble:"

Each relation in a relational model represents a fundamental aspect of intension in the modeler's depiction of reality. Association and the use of relational operators effect that fundamental intension deriving an answer to any query we may invent based on that fundamental knowledge. The result of every relational operation is itself a relation. The modeler's ingenuity and discipline in forming queries carefully that yield results, relations, that are themselves consistent with the integrity constraints of the model creates the potential of an endless cascade of query result as input to another query and so on. This is the direct result of the mathematical formalism upon which the relation model is based –the predominating strength of the relational paradigm. The form in which these queries may be posed to a relational system is constrained only by the choice of mathematical representations (e.g. tuple calculus or domain calculus) or transformations (e.g. relational algebra or relational calculus) to the underlying relational definition.

9. IDENTITY reliant upon Encapsulation, Modularization, Composition of Function, Scale, Programmability and Elegance *(derived from Alexander's Contrast reliant upon Boundaries, Positive Space, Deep Interlock and Ambiguity, Gradients, The Void and Not Separateness)* incorporated in the choice by the modeling action "to identify:"

Identity is at the root of recognition. In the physical world identity is literal based upon direct sensorimotor experience: by sight or touch and in some cases by sound or smell – a human experience of the "real" world. In the relational paradigm this human experience is applied directly by collecting those attributes that completely describe how any particular instance is unique – the combination of attributes that comprise the primary key. The primary key serves to anchor the knowledge that surrounds it – those additional attributes that further describe the tuple which it uniquely determines –those attribute values that are functionally dependent upon the primary key. No tuple is permitted to exist in the relational universe (extension) unless it has a primary key – entity integrity. Ownership as it is manifest through foreign key associations is anchored on the primary key of the owner tuple.

10. SCALE reliant upon Stepwise Refinement, Cohesion, Transparency, Identity, User Friendliness, Patterns and Elegance *(derived from Alexander's Gradients reliant upon Levels of Scale, Strong Centers, Local Symmetries, Contrast, Roughness, Echoes and Not Separateness)* incorporated in the choice by the modeling action "to focus:"

In many cases the only familiarity that is needed in a relational model is the intension – the collection of relation definitions with their attribute sets defined by their respective attribute domains and the associations among the relations. The knowledge structure and semantic relationships that may be mined through relational operators sufficiently defines any derivation of information representations that queries may be formulated to elicit. In terms of scale any relational model (intension or extension) may be expanded to incorporate additional knowledge. The modeler achieves this by grafting new knowledge onto existing relation structure through the addition and/or alignment of data attribute domains and associations.

11. USER FRIENDLINESS reliant upon Cohesion, Modularization, Correctness, Scale, User Friendliness, Reliability and Elegance *(derived from Alexander's Roughness reliant upon Strong Centers, Positive Space, Good Shape, Gradients, Roughness, Simplicity and Inner Calm and Not Separateness)* incorporated in the choice by the modeling action "to accommodate:"

There is elegance in the succinctness and simplicity that arises from properly isolating domain knowledge in the respective relations. The use of user / client / customer familiar naming of relations and attributes and the choice of the commonly used, domain based attribute values lends a comfort level to the representation of problem domain experience. The relational model also enables the derivation of contained knowledge at levels of granularity much higher than the individual tuple or relation. This is because relational operations on relations produce relations as their result. Information derived from a relational database can be presented as if it were simply retrieved from a single physical relation. This illusion is easily achieved in relational programming languages that support the definition and storage of queries that may then be referenced themselves as relations without the users' notice (i.e. in ANSI SQL the "create view" syntax.). The facility of such extensions to apply relational operations so discretely creates virtually unlimited opportunities and permits what might otherwise be a complex and daunting algorithm of derivation to be completely ignored by the users.

12. PATTERNS reliant upon Stepwise Refinement, Correctness, Transparency, Scale, User Friendliness and Elegance *(derived from Alexander's Echoes reliant upon Levels of Scale, Good Shape, Local Symmetries, Gradients, Roughness and Not Separateness)* incorporated in the choice by the modeling action "to pattern:"

There is elegance in the succinctness and simplicity that arises from properly isolating domain knowledge in the respective relations. The use of user / client / customer familiar naming of relations and attributes and the choice of the commonly used, domain based attribute values lends a comfort level to the representation of problem domain experience. The relational model also enables the derivation of contained knowledge at levels of granularity much higher than the individual tuple or relation. This is because relational operations on relations produce relations as their result. Information derived from a relational database can be presented as if it were simply retrieved from a single physical relation. This illusion is easily achieved in relational programming languages that support the definition and storage of queries that may then be referenced themselves as relations without the users' notice (i.e. in ANSI SQL the "create view" syntax.). The facility of such extensions to apply relational operations so discretely creates virtually unlimited opportunities and permits what might otherwise be a complex and daunting algorithm of derivation to be completely ignored by the users.

13. PROGRAMMABILITY reliant upon Stepwise Refinement, Encapsulation, Modularization, Transparency, Identity and Reliability *(derived from Alexander's The Void reliant upon Levels of Scale, Boundaries, Positive Space, Local Symmetries, Contrast and Simplicity and Inner Calm)* incorporated in the choice by the modeling action "to generalize:"

Returning again to the use of relational operations to compose higher and higher levels of information we see individual relations as building blocks that may be arranged (assembled through relational operations) to yield any reasonable arrangement or derivation of information that the underlying relations may possess. This is possible because of the individual identity that each relation fosters in its tuples and because of the predictable reliability that proceeds from the consistency and safety of relational operations that is guaranteed in a set of normalized relations. The extent of information mining that may be attempted is limited almost solely by the programmers' imagination.

14. RELIABILITY reliant upon Transparency, Composition of Function, Patterns, Programmability, and Elegance *(derived from Alexander's Simplicity and Inner Calm reliant upon Local Symmetries, Deep Interlock and Ambiguity, Echoes, The Void and Not Separateness)* incorporated in the choice by the modeling action "to normalize:"

There is an overarching simplicity that results from the fact that all of the properties of integrity are based upon data attribute values that may be readily inspected before or after any relational operation. Intension is expressed in modeled expressions of integrity constraints that are domain specific. The synchronization between the intension and extension of the model is easily tested because of this simple transparency. Reliability is assured if valid relational operations are applied consistent with model integrity constraints and thus will always yield consistent ("truthful") information.

15. ELEGANCE reliant upon Encapsulation, Modularization, Composition of Function, Scale, User Friendliness, Programmability and Reliability ***(derived from Alexander's Not Separateness reliant upon Boundaries, Positive Space, Deep Interlock and Ambiguity, Gradients, Roughness, The Void and Simplicity and Inner Calm)*** **incorporated in the choice by the modeling action "to coordinate:"**

Elegance is achieved largely through the relational model when relations are modeled with a minimum of extraneous or redundant information. Indeed eliminating redundancy is common mantra of relational modeling. The laying out of basic facts divided into distinct encapsulated containers of knowledge and the subsequent composition of higher levels of derived information effects a sense of economy of form and abundant opportunity for exploring and extracting the knowledge a database so fashioned accommodates.

Programming Languages vs. Ontology

The relational ontology may be applied supporting the fifteen properties of choice derived from Christopher Alexander's theory of wholeness and life. However only the integrity properties and normalization hint at the range of transactions that a relational model is usually composed to support. The absence of this explicit behavioral information restricts the modeler's ability to expose the elements of system development distinguished by Fred Brooks' as essence and accidents of implementation. The focus on data in the relational model to the effective exclusion of behavior results in models that predominate in their depiction of essence from the problem domain; but an essence that is almost entirely static in nature. The designation of primary and foreign keys and the process of normalization that imprints semantics on the static data structures only hints at the eventual range of behavior that will ensue in the operational descendent of the model, the physical relational database.

There may be many variations of language features implementing the relational paradigm that will influence the representation of the properties described in this chapter and the current generation of languages will surely be augmented by many to come. However, any language implementation that supports the rubrics of the relational ontology described here can contribute the achievement of these fifteen properties in the relational paradigm of representing data-focused domain knowledge.

Bibliography

Alexander, Christopher, Ishikawa, Sara, Silverstein, Murray, Jacobson, Max, Fiksdahl-King, Ingrid and Angel, Shlomo, A Pattern Language, New York, NY: Oxford University Press, 1977.

Alexander, Christopher, The Nature of Order An Essay on the Art of Building and the Nature of the Universe: Book I - The Phenomenon of Life, Berkeley, California: The Center for Environmental Structure, 2002.

Alexander, Christopher, The Nature of Order An Essay on the Art of Building and the Nature of the Universe: Book II - The Process of Creating Life, Berkeley, California: The Center for Environmental Structure, 2002.

Alexander, Christopher, The Nature of Order An Essay on the Art of Building and the Nature of the Universe: Book IV - The Luminous Ground, Berkeley, California: The Center for Environmental Structure, 2004.

Alexander, Christopher, The Nature of Order An Essay on the Art of Building and the Nature of the Universe: Book III - A Vision of a Living World, Berkeley, California: The Center for Environmental Structure, 2005.

Codd, E.F., "Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks", IBM Research Report, 1969.

Codd, E.F.,"A Relational Model of Data for Large Shared Data Banks", *Communications of the ACM*, 13, 6 [June 1970]: 377-387.

Waguespack, Leslie J. Jr., "Hammers, Nails, Windows, Doors and Teaching Great Design." In *Proceedings of ISECON*, (Pittsburgh, PA, USA), November 2007.

Waguespack, Leslie J. Jr., Thriving Systems Theory and Metaphor-Driven Modeling, Technical Report, Department of Computer Information Systems, Bentley University, Waltham, MA, 02154, 2009.

