Published in S. Hirtle, A. Frank (Eds.), Spatial Information Theory: A Theoretical Basis for GIS, Proceedings of the International Conference COSIT '97, Lecture Notes in Computer Science 1329, Springer-Verlag, pp. 15-33.

Qualitative Representation of Change*

Kathleen Hornsby and Max J. Egenhofer
National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
University of Maine
Orono, ME 04469-5711, U.S.A.
{khornsby, max}@spatial.maine.edu

Abstract. Current geographic information systems (GISs) have been designed for querying and maintaining static databases representing static phenomena and give little support to those users who wish to represent dynamic information or incorporate temporality into their studies. In order to integrate phenomena that change over space and time in GISs, a better understanding of the underlying components of change and how people reason about change is needed. This paper focuses on a qualitative representation of change. It offers a classification of change based on object identity and the set of operations that either preserve or change identity. These operations can be applied to single or composite objects and combined to express the semantics of sequences of change. An iconic, visual language is developed to represent the various types of change and applied to examples to illustrate the application of this language. Such a formalization of the basic components of change lays the foundation for a new generation of formal models that captures the semantics of change and leads to improved interoperability between GISs and process models or simulation software.

1 Introduction

Scientists from many disciplines have an interest in representing dynamic phenomena. Epidemiologists simulate the spread of disease in different environments searching for clues in the pattern of disease occurrence that will aid in the prevention of further spread of the illness (Cliff *et al.* 1981; Cliff *et al.* 1992; Ackerman 1994) , while coastal geomorphologists are interested in describing the materials and processes that affect coastal form (Raper and Livingstone 1995), and wildfire modelers study the characteristics of the growth and spread of wildfires (Clarke *et al.* 1994; Yuan 1994; Xu and Lathrop 1995) . These researchers strive to understand change in order to improve their ability to make accurate predictions of what the state of an entity will be at a future time. They require tools that will aid them in their explanations of why

---

* This work was partially supported by the National Science Foundation for the National Center for Geographic Information and Analysis under NSF grant number SBR–8810917. Max Egenhofer's research is further supported through NSF grants SBR-8810917, IRI-9309230, IRI-9613646, SBR-9700465, and BDI-9723873; grants from Rome Laboratory under grant number F30602-95-1-0042 and the National Imagery and Mapping Agency under grant number NMA202-97-1-1023; and a Massive Digital Data Systems contract sponsored by the Advanced Research and Development Committee of the Community Management Staff.

a certain state exists–either now or in the past–and assist them with proper and efficient resource allocation (Shoham 1988) . Current geographic information systems (GIS) have only limited ability to represent phenomena that change over space and time. While they can provide snapshot views at discrete time intervals (DiBiase *et al.* 1992; Chrisman 1997) , they fall short in providing an ability to link the process models of scientists with data from multiple sources or simulate scenarios of change for users.

Change happens over space and time. Although methods for integrating temporal change into GISs have been the focus of much research (Armstrong 1988; Al-Taha and Barrera 1990; Langran 1992; Frank 1994; Peuquet, 1994; Worboys, 1994; Claramunt and Thériault 1995; Claramunt and Thériault 1996; Egenhofer and Golledge 1997), current systems lack an ability to represent temporality. Al-Taha and Barrera (1990) recognized two methods for reasoning about time. A *time-based* approach considers time as a separate dimension and uses points or intervals as primitives, while a *change-based* approach concentrates on recording changes or facts that are valid at a certain time, without considering explicitly the temporal domain. Further work in the development of temporal GISs from a change-based perspective includes studies of the mechanisms necessary to model the evolution of feature identities through time (Al-Taha and Barrera 1994) and the use of direct-manipulation interfaces for natural process modeling (Sleezer 1994). Claramunt and Thériault (1995, 1996) introduced a framework for modeling the semantics of spatio-temporal processes and described certain basic events and processes using formal methods. The present investigation builds upon these concepts and considers further this change-based approach to data modeling, focusing on the underlying components of change and incorporating how humans reason about change over space and time. The work is based on the concept of object identity, a fundamental element in object-oriented programming and object-oriented database systems (Khoshafian and Copeland 1986). In this paper, change refers to those operations that may be performed on an object or set of objects and either preserve object identity or result in a change of identity. An iconic, visual language is used to express these operations. This paper describes a classification and representation of different types of change beginning with changes to single objects and building to more complex scenarios of change.

Section 2 introduces the concepts of objects and object identity and describes identity with respect to its usage in programming and database languages. A framework for representing types of change using a visual language is introduced in Section 3, with manipulations of single and composite objects and a classification of changes resulting from the joining and splitting of these objects. Section 4 describes how relations are added to the framework and Section 5 introduces operations on the properties of objects and how they may be transferred between objects. The visual language is used to model different views of change (Section 6). Conclusions and future work are discussed in Section 7.

2 Objects and Object Identity

Object is the term used to represent in an information system a real world “thing” that might exist as a physical entity, such as an island, a building, or a mountain, or something conceptual such as the State of Maine or the North Sea (Smith 1995) . An object may represent a mapped feature like a lake or a land parcel, or be something that is not readily mapped such as a salt dome (Laurini and Thompson 1992) . Objects are not necessarily homogeneous, but may be composed of more than one part and contain other objects.

In programming or database languages, the concept of a unique object *identity* is commonplace (Khoshafian and Copeland 1986) . Object identity has been defined as the trait that distinguishes an object from all others (Khoshafian and Baker 1996) . Identity provides a way to represent the individuality or uniqueness of an object, independent of its attributes and values. This concept aids the idea of an object being a stable and enduring element, something on which we can have a perspective (Smith 1996) . This identity may be implemented at the system level by ensuring that each system object has a unique identifier, created when the object is created, never altered, and only removed once the object has been destroyed (Worboys 1994) . Unlike an object's *identity,* which represents an object's uniqueness, its *properties* or characteristics may be shared with other objects. When scenarios of change are under consideration, identity plays an important role in helping to keep track of alterations to objects and assists in determining the existence or non-existence of objects.

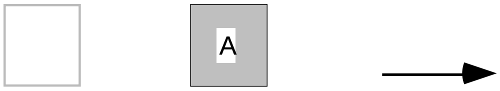
Every programming and database language must have some way to tell one object from another. Although many languages utilize user-defined variable names to distinguish objects, this alone is not considered strong enough support for identity, mixing concepts of addressability with identity (Khoshafian and Copeland 1986) . The purpose of addressability is to provide the means to access an object within a particular environment and, therefore, is not strictly a characteristic of an object. Identity, however, is specific to an object. Its purpose is to provide a way to represent the individuality of an object independently of how it is accessed. Address-based identity mechanisms, therefore, are considered to compromise identity when ideally the language should provide separate mechanisms for the two concepts.

The use of identifier keys (i.e., attributes that uniquely identify a tuple) to distinguish objects as practiced in current GISs and database systems is not completely satisfactory as it confuses issues of data value with identity (Khoshafian and Copeland 1986; Bonfatti and Pazzi 1995; Khoshafian and Baker 1996) . One problem with the identifier keys approach is that these keys cannot be changed. For example, a college name in a university may be used as the identifier key for that college, but if the name has to change under a reorganization, there will be a discontinuity in identity for the college as well as update problems in all objects that refer to it. Also, identifier keys may not be able to provide identity for every object in the data model. Each attribute or useful set of attributes cannot have an identity. The choice of which attribute to use for an identifier key may need to change. If two databases are required to merge, merging the identifier keys may be a problem if the keys are of different types or different combinations of attributes.

Offering better support for identity, object-oriented languages employ separate mechanisms for these concepts so that each object maintains a unique and consistent notion of identity, regardless of how it is accessed or how it is modeled with descriptive data. The object is an instance of a class (its type) and each object has a state, which are the values of its instance variables (Kim 1995; Khoshafian and Baker 1996) . In addition, each object has a built-in identity, which is independent of its class or state. The identity of an object is generated by the system when the object is created. Object-oriented systems supporting strong built-in identity also allow the object to undergo structural modification (i.e., changing its class) without changing its identity.

3 Object Identity Operations

This investigation focuses on developing a framework that classifies types of changes applicable to objects and their identities, and in particular captures the richly varying semantics of change. The framework is developed using a Change Description Language (CDL), a visual language based on an iconic representation of different kinds of change. Visual languages rely on a combination of network, topological, and metrical relationships to express semantics (Helm *et al.* 1991) . They offer users an alternative means of communication with a computer system, which is often easier and clearer than standard SQL input (Catarci *et al.* 1993) . The CDL uses basic symbols for object non-existence (Figure 1a) and object existence (Figure 1b). These symbols represent states in which objects reside at different times. The transition from one state to the next is captured through an arrow (Figure 1c). Objects have been given a label to aid identification of unique identities.



(a) (b) (c)

Figure 1: Basic symbols used for (a) object non-existence, (b) object existence, and (c) transition.

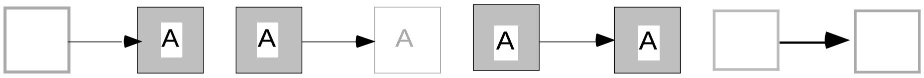
Temporal change is represented qualitatively based on the temporal order of transitions, an approach to temporal reasoning that has been found to be acceptable to many of the domains using GIS (Frank 1994) . Scenarios are developed from left to right, where “left” corresponds to “before” and “right” to “after.” Temporal concurrence of transitions affecting different objects is shown by aligning these states in the vertical. No quantitative measures are represented and though concurrent transitions can be depicted, no information on the *duration* of a transition is assumed.

3.1 Manipulating Single Objects

Through combinations of the basic symbols, a set of operations is introduced to manipulate individual objects. A similar set of operations with respect to tracking the evolution of a temporal feature’s identity was identified by Al-Taha and Barrera (1994), and Claramunt and Thériault (1996) described basic changes of appearance, disappearance, and stability in their work on the representation of spatio-temporal processes. This work provides a more comprehensive study of identity-based change operations and how these operations can be combined to reflect different types of change.

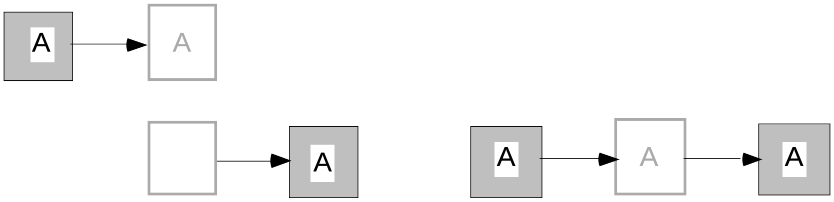
The operation `create` describes the formation of an object with its identity where no object existed previously (Figure 2a). `destruct` is the inverse to `create` as it eliminates an object that was previously created (Figure 2b). The operation `continue` refers to the continued existence of an object and its identity from one transition to the next (Figure 2c). It is also possible under certain circumstances to have a transition where the object is not in existence (Figure 2d).

This set of basic operations serves as the foundation for a change-based representation of spatial phenomena. These operations can be further combined to describe scenarios of change. For instance, Figure 3a shows the destruction of one object and the subsequent creation of a new object with a different identity. Although this object has the same label, it is not the same object as before. This view is semantically different from sequences of operations in which an object is destroyed and subsequently recreated (Figure 3b). In the latter case, the operations are applied to the same identity, capturing the semantics of a *reincarnation*. Therefore, `reincarnate` may be defined as a `destruct` operation followed by a `create` of an object with the same identity.



(a) (b) (c) (d)

Figure 2: Identity operations on single objects: (a) `create`, (b) `destruct`, (c) `continue existence`, and (d) `continue non-existence`.



(a) (b)

Figure 3: Two semantically distinct combinations of create and destruct: (a) `destruct` operation applied to one object followed by a `create` operation forming an object with a different identity and (b) `reincarnate`.

In addition to a `create`, objects may be subject to other changes when a new object is formed from one that existed previously. This type of transition, the *issuing* of a new object from an existing object (Figure 4), is found in many types of change.

Figure 4: *Issue* is a special type of transition.

One type of change involving *issue* is where an object `spawns` an object (Figure 5a) with a new identity that is unique and separate from that of the original object. The original object and identity continue to exist. For instance, at the time India gained independence from Britain in 1947, East and West Pakistan were spawned, while India's identity continued to exist. Another type of change, exemplified by the operation `metamorphose`, also results in the issuing of a new object, but in this case, the original object ceases to exist (Figure 5b). Examples of this type of operation come from such changes as when a landfill site is reclaimed and becomes a golf course, or a wilderness area is developed into a housing estate.

3.2 Combining Objects

Change also occurs when two objects join together and form an object with a new identity. Figure 6 presents three different classes of joins (not to be confused with the join in relational algebra) between single objects. Although these operations are depicted between two objects, clearly they may involve more objects. These joins are expressed in terms of the transition *issue* and identity operations, `destruct` and `continue.`

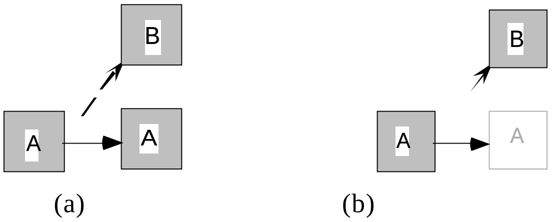


Figure 5: Operations involving the transition type *issue*: (a) `spawn` and (b) `metamorphose`.

In the case of a `merge`, the original objects are destroyed at the time of the merge and an object with a new identity is issued (Figure 6a). This type of change is common when dealing with change of ownership of land parcels or with the amalgamation of properties, such as farms. It is necessary to preserve a relationship between the successor object and its predecessors, since the successor is formed as a result of the `merge` and is semantically different from a `create` where no object existed previously. It is also possible for two or more objects to join such that they `spawn` a new object (`generate`), and continue to exist (Figure 6b). Parenthood is the classic example of this type of issuance. The third type of combination, `mix`, refers to those cases where one of the parent objects is destroyed in the join (Figure 6c), but not both as with a `merge`. A new object and identity are issued as a result of this operation.

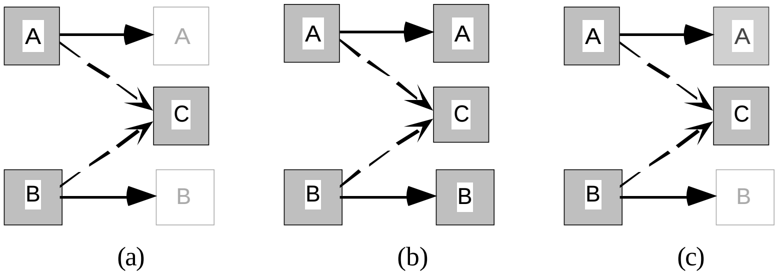


Figure 6: Combining single objects: (a) `merge`, (b) `generate`, and (c) `mix`.

Joins can also result in the formation of composite objects which may be composed of 2...*n* subparts and may be nested (Figure 7). The creation of composite objects is a fundamental abstraction method in object-orientation, commonly referred to as aggregation (Smith and Smith 1977; Kim *et al.* 1987; Schiel 1989). These structures capture the notion than an object may be composed of other objects, representing a hierarchy of objects. Although the CDL representation of the composite shows A and B to be disjoint, this is the default representation and is not meant to confer any detail regarding the spatial relation between two objects in the composite.

Figure 8 shows five types of join involving composite objects. With an `aggregate` operation, two or more single objects join to form a composite. The identities of the original objects are maintained and continue to exist within the newly formed composite (Figure 8a). A new identity is created for the resulting composite object. This operation is familiar in the case of regions aggregating to form a larger political unit, such as a nation.

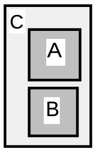
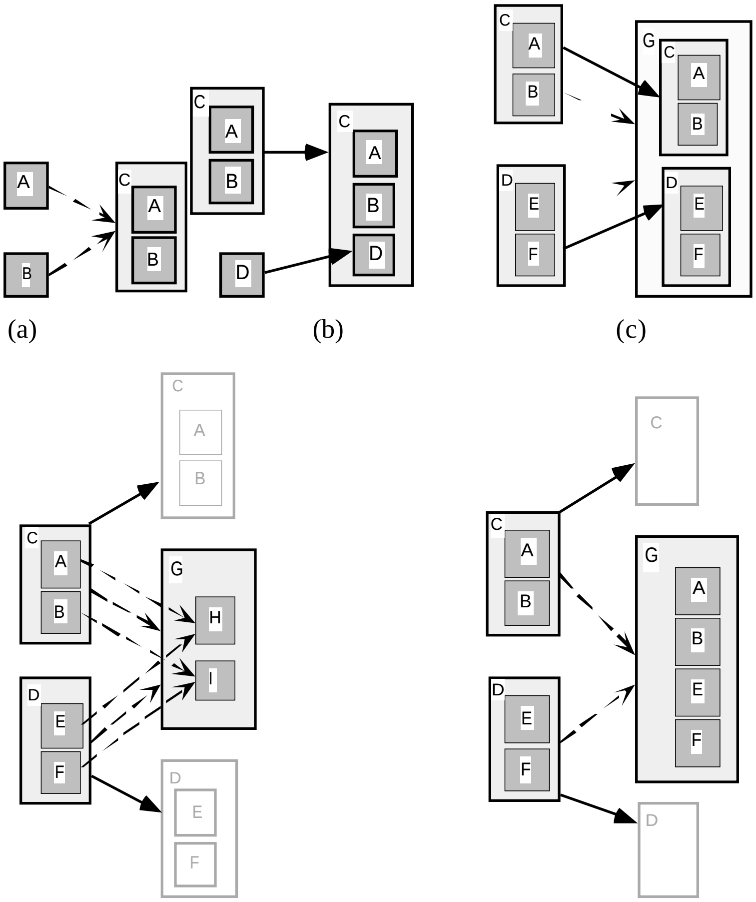


Figure 7: Composite object C with subparts A and B.



(a) (b) (c)

(d) (e)

Figure 8: Forming composite objects: (a) `aggregate`, (b) `compound`, (c) `unite`, (d) `amalgamate`, and (e) `combine`.

Once the composite has been formed, a `compound` operation describes the situation where more objects are added to the composite as subparts (Figure 8b). The composite identity continues to exist as these additional joins do not result in the creation of new composite identities. A familiar example of the `compound` operation is that of individual states joining the Union. Each state has its own identity and is a subpart of the Union that can be viewed as a composite object.

It is also possible for a join to be performed between composite objects (`unite` operation) (Figure 8c). A new composite object and identity results from the join while the original objects continue as subparts of the new composite. For instance, two towns that are considered as composite objects may unite to form a school administrative district. This is viewed as an operation distinct from `aggregate` because it requires two or more composites, not individuals, to be joined. `unite` and `aggregate` may be combined into a higher-level operation if overloading of the operators is allowed (Khoshafian and Abnous 1995).

Two composite objects may also `amalgamate` to form a new composite (Figure 8d). The resulting object will contain subparts that are actually a `merge` of the original subparts. The composite objects including all identities of the subparts are destroyed in this join. As an example, if two metropolitan areas join together to form a new, larger unit, subparts such as local governments, will not be duplicated in the resulting amalgamation, but will merge to form a new subpart.

Finally, the `combine` operation describes the change that occurs when two composite objects join to form a new object, and the original composite objects are destroyed in the join (Figure 8e). The individual identities of the subparts continue to exist. Here, for instance, the case of two administrative units joining to form a larger unit and losing the aggregate level, with the subparts continuing to exist serves as an example.

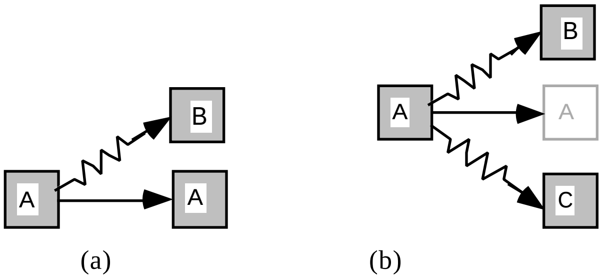
3.3 Splitting Objects

Change also takes place through the splitting of objects. This type of transition is referred to as *separate* (Figure 9). Although a new object is formed from an existing object as with an *issue*, this type of transition is semantically different from *issue*, since the resulting object derives from the original object and, therefore, the original is diminished by this operation.

Figure 9: The transition type *separate*.

Figure 10 shows the different kinds of split operations possible on single objects. In the first case, `splinter`, a portion of the object separates from the original (Figure 10a). The original object and identity continue to exist. This operation is exemplified by the process of erosion as encountered in the physical world (Armstrong 1988), or the formation of splinter groups as a social construct.

In the second case, `divide`, the original object separates into *n* parts (Figure 10b). The original object ceases to exist. Each successor object has its own unique identity. Note that this is *not* the inverse operation of the `merge` operation. Once a single object has been split, the original identity is destroyed and generally, will not be created again (unless there is a reincarnation process). Some relationship exists between the survivor objects and their predecessors.



(a) (b)

Figure 10: Splitting single objects: (a) `splinter` and (b) `divide`.

It is also possible to split composite objects. In a `secede`, one or more components of the original may split off along explicit boundaries (Figure 11a). The identity of the separated object(s) continues to exist. The identity of the original composite object also continues. This type of operation is exemplified by a region separating from its mother country.

Another type of split from a composite is the `dissolve` operation where the composite completely splits into its *n* parts (Figure 11b). The composite object and its identity now no longer exists, while the subparts continue. This is the inverse of the `aggregate` operation. A nation may dissolve into separate parts, such as the Soviet Union's dissolution into regions. A `dissolve` operation may be triggered by a `secede` operation. For instance, if the composite object is dependent on its components for existence and a `secede` operation occurs, the composite identity may no longer be able to exist independently and will dissolve.

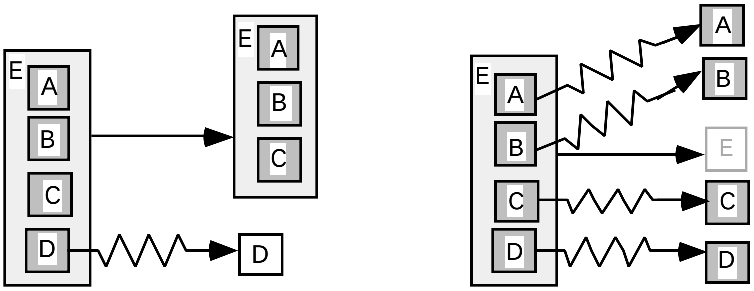


Figure 11: Splitting composite objects: (a) `secede` and (b) `dissolve`.

3.4 Selecting an Object

With an operation such as `secede` it is necessary to have a `select` operator that first allows for choice or selection based on some defined criteria of either the entire object or a portion of the object (Figure 12). The `select` operator is also illustrated in Figure 11a as part of the `secede` operation.

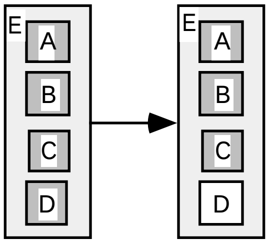


Figure 12: `select` operator applies a selection to subpart D of composite E.

3.5 Examples of Combining and Splitting Objects

The following examples illustrate how the CDL assists our ability to represent different types of change. Consider as a first example, the founding of the Dominion of Canada.

Founding the Dominion of Canada.

In 1867, four provinces (Ontario, Quebec, New Brunswick, and Nova Scotia) joined together to form the new nation, Canada. The provinces may be considered as objects with their own unique identities. The object Canada, formed as a result of the `aggregate` operation, is a composite object, also with its own identity (Figure 13). Note that this example does not consider any information about spatial properties or spatial relations. The provinces are depicted in alphabetical order with no attempt to represent information about their spatial properties such as neighborhood, orientation, or size.

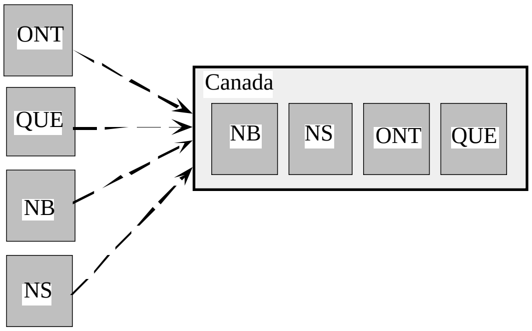


Figure 13: Forming Canada in 1867 with provinces Ontario (ONT), Quebec (QUE), New Brunswick (NB), and Nova Scotia (NS).

Over time, modeled through successive `compound` operations, more provinces join Canada (Figure 14). As each province joins the Dominion, the identity of the object Canada continues while each province maintains its unique identity. A simplified view of present-day Canada is shown, depicting provinces only (Figure 15).

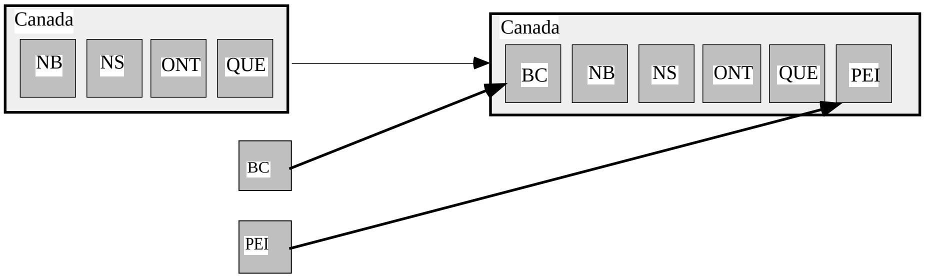


Figure 14: British Columbia (BC) and Prince Edward Island (PEI) join Canada.

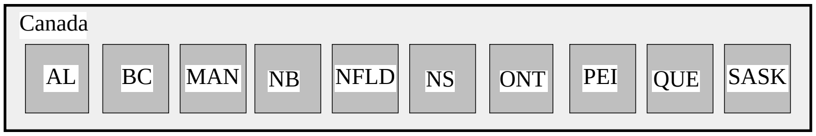


Figure 15: Composite object representing current provinces that comprise Canada: Alberta (AL), British Columbia (BC), Manitoba (MAN), New Brunswick (NB), Newfoundland (NFLD), Nova Scotia (NS), Ontario (ONT), Prince Edward Island (PEI), Quebec (QUE), Saskatchewan (SASK).

Hypothetical Separation of Quebec from Canada.

If Quebec were to separate from Canada, the identity of Quebec would continue to exist although it is no longer a subpart of Canada. Canada, as a composite object with an identity, continues to exist. This hypothetical scenario is modeled through the application of the `select` operator (Figure 16) followed by a `secede` operation (Figure 17), resulting in both a composite object and a single object.

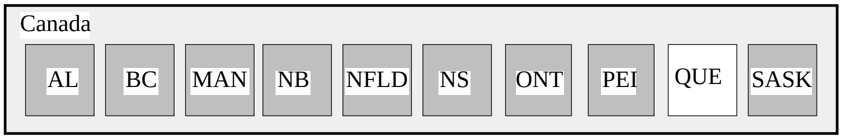


Figure 16: `select` operator is used to choose Quebec.

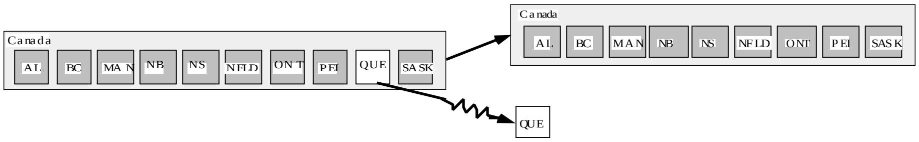


Figure 17: Secede operation performed on object Canada.

This change may also be viewed from a different perspective than the secede shown above. If Quebec were to separate from Canada, it might be replaced by a new object, “Nouveau Quebec” (Figure 18).

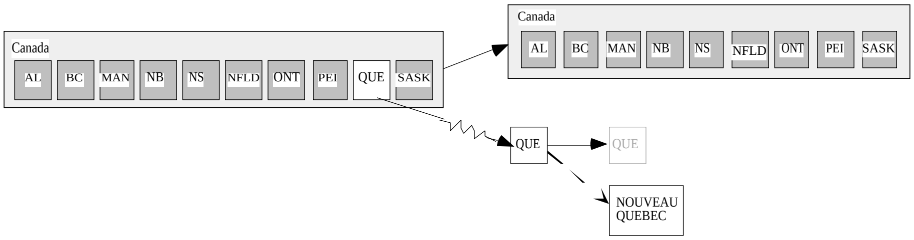


Figure 18: A new object, "Nouveau Quebec," is formed after separation, while the composite object Canada continues and Quebec is destroyed.

4 Relations among Objects

So far the visual language has been capable of expressing join and split operations for both single and composite objects, for showing combinations of these operations, and for representing more than one view of the same transition. Reasoning about change is enhanced if we incorporate information concerning *relations* between objects. A relation (or set of relations) represents some association or condition among objects (Figure 19).

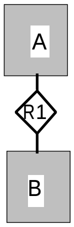


Figure 19: Relation R1 exists between objects A and B.

This association may include topological relations that hold under such transformations as rotation, scaling, and translation, including the recognized topological relations of meet, disjoint, contains, etc., (Egenhofer and Herring 1990) or other spatial relations such as orientation (Frank 1996). The relation could also represent a link or connection–either physical or semantic–between objects.

Relations may be added, removed, or changed. Relation R1 is established between objects A and B for the `add relation` operation (Figure 20a). Similarly, `remove relation` is where Relation R1 between A and B is no longer present (Figure 20b), and `change relation` describes the situation where Relation R1 is changed to R2 (Figure 20c). These operations involving relations do not change the identity of the objects. Identity continues in all cases.

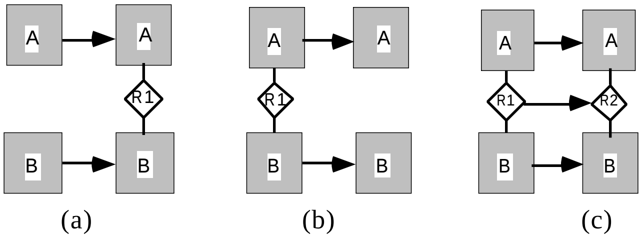


Figure 20: Relations among objects: (a) `add relation`, (b) `remove relation`, and (c) `change relation`.

5 Properties of Objects

Similar to the definition used in the Entity-Relationship Model (Chen 1976) , we define for every object a corresponding set of *properties,* that describe the object (Figure 21). These properties are values that may be obtained through observation or measurement. They may be grouped into value sets and are typically classified as *geometric,* such as shape, size, dimension, orientation, or location, and *non-geometric,* such as color, or name of object. In this paper we will treat properties in a generic fashion, not distinguishing different types of properties.

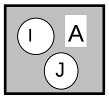


Figure 21: Object A has properties I and J.

The location of the property within the object is immaterial (Figure 22a). When an object continues to exist, the properties also continue (Figure 22b), whereas when an object is destroyed, all of its properties are destroyed as well (Figure 22c).

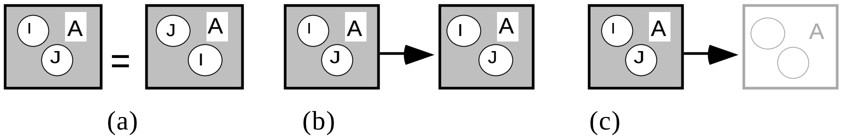


Figure 22: (a) Location of property within an object is immaterial, (b) properties continue as object continues, and (c) when the object is destroyed, properties are destroyed.

Properties may be added to an object, or removed from an object. Properties are added to an object after it has been created or issued (Figure 23a). Properties can only be removed from an existing object (Figure 23b). Identity is not affected by the addition or removal of properties.

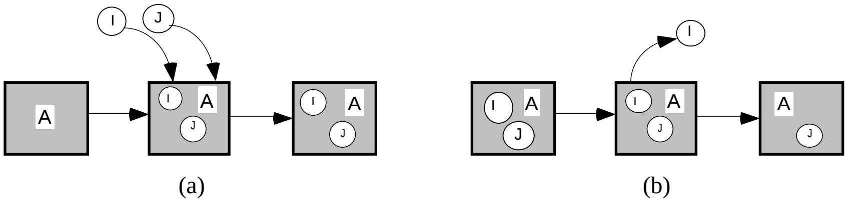


Figure 23: Adding and removing properties of objects: (a) `add property` and (b) `remove property`.

As an example, imagine the addition of a swimming pool to a backyard. A real estate agent may view the pool as a property of the backyard–an asset–for when the owner wants to sell the house. Adding a pool to a backyard can thus be viewed as adding a property (Figure 24). The identity of object "backyard" does not change with the addition of the new property "pool."

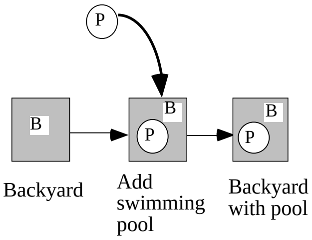


Figure 24: Adding a property (pool) to an object (backyard).

Additional operations on properties involve a transfer of properties from one object to another. Claramunt and Thériault (1996) introduced the notion of succession as an event that occurs when an object Y is the immediate successor of an object X. It is also possible to transfer properties in different ways. The `pass on` operation describes the situation where an object can transfer one or more properties while continuing to retain those properties (Figure 25a), or in the case of a `succeed`, a property can be transferred, but upon doing so the property is removed from the object (Figure 25b).

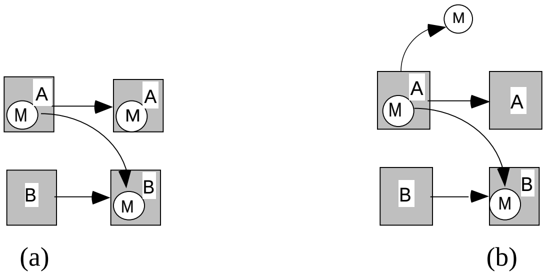


Figure 25: Transfer operations on properties: (a) `pass on` and (b) `succeed`.

Other types of transfers of properties can be expressed through a combination of these operations. For instance, an `exchange` of properties between objects can be expressed through a combination of `pass on` and `remove` operations (Figure 26). In all cases, object identity continues unchanged.

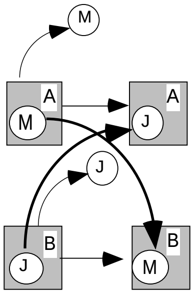


Figure 26: Exchange properties M and J between objects A and B.

6 Applying the Change Description Language

The CDL can be used to express the components and transitions that comprise different scenarios of change. Many of the examples presented in this paper show that it may be necessary to combine operations in order to model the set of criteria that a particular sequence of change requires. Consider the spread of measles, an example of contagious diffusion, which involves changes in the state of health from person to person. If a person infected with measles comes into contact with a susceptible person, this usually means that infection will occur in the susceptible case. One way these stages can be shown is by representing each individual as an object and showing the infection being passed on as a property (Figure 27). In this simplified example, three individuals (A, B, and C) are susceptible to infection while the fourth (D) gains early immunity from vaccination. Their subsequent changes in health are shown as a sequence of transitions, and concurrent transitions are depicted by aligning the objects vertically. With the addition of the property measles, one person becomes infected and upon contact with another individual (shown as relation “contact” between two objects), passes on the infection. Change takes place when an object or property is affected by the spread phenomenon (i.e., a susceptible individual becomes infected). No change to the individual occurs, however, when the immune individual has contact with the infection.

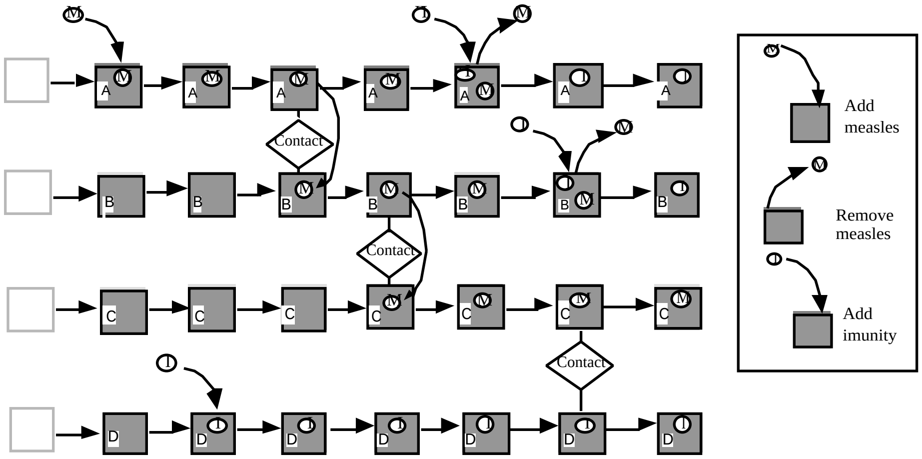


Figure 27: Spread of measles between four individuals (A, B, C, D).

The depiction of spread in this example may differ from that held by a microbiologist who views this virus as an object rather than a property. In this case, the virus is an object that spawns other virus-objects before becoming extinct. The new viruses in turn spawn more and the cycle continues as long as there are sufficient hosts. The life of the virus-object is modeled through the operations of `create`, `spawn`, and `destruct`.

The CDL can also be applied to describe the components of fire chemistry in order to understand wildfires, another example of a spreading phenomenon (Clarke *et al.* 1994). The life of a wildfire episode may be modeled as a `create` that describes the ignition of the fire by lightning for instance, followed by numerous instances of `spawn` and periods of `continue` of object existence (representing the propagation and continuation of the fire) and followed finally by `destruct` (extinction phase) when the fire is doused by rain. These examples highlight the fact that humans reason about change in different ways and from different perspectives. Hence the models designed to represent scenarios of change must be able to accommodate these differences and capture the semantics of the transitions under consideration.

7 Conclusions and Future Work

A framework for representing the semantics of different kinds of change and a classification of operations that result in preservation or change to object identity has been introduced. The model is extended in a stepwise fashion through combining change operations to capture more semantics. In addition to the operations, two special types of transition, *issue* and *separate*, are defined in order to distinguish the formation of objects under particular circumstances. The change-based representation is extended through the addition of relations between objects, and inclusion of operations involving properties of objects. The visual CDL developed to express combinations of the change operations is flexible enough to represent different user views of change and makes it possible to model sequences of change. The set of change operations was derived from describing different types of join and split operations, and a systematic derivation of all possible combinations is under investigation.

The CDL has represented certain kinds of change in a simple, easy-to-understand fashion. Yet, this approach does have its limitations. The language becomes cluttered and difficult to decipher especially when dealing with changes involving composite objects. Also, it is very difficult to employ a graphic representation when describing any change that involves spatial properties of an object, or spatial relations among objects. If Figure 24, the pool in the backyard, is considered again, this time from an object view, and now incorporating information on the geometric shape of objects (e.g., oval pool in rectangular backyard), a user may be uncertain whether metric information is also to be inferred (Figure 28). For instance, is the pool really closer to one end of the backyard than the other, or is this just an artifact of the iconic representation?

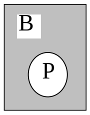


Figure 28: Oval pool in rectangular backyard.

As another example, consider the spatial relation, *meet*, which describes the situation where two objects share a common boundary. Using the iconic language, it is not possible to represent the meet relations between say, New England states in the U.S., unless information on the orientation of these objects is also available. In these examples, each additional piece of information brings us closer to a map-like representation. Because of these limitations, the rules for change as developed with the CDL will be translated into a symbolic form for further extensions to the model.

The important aspect of granularity and its effect on how change is viewed will be another focus to future work. Different scenarios of change require different levels of abstraction, and different rules to govern dependencies between objects, their properties, and relations with other objects. Maintaining the flexibility to represent different levels of granularity that are in keeping with user views will be important to future efforts in extending the model.

8 Acknowledgments

The authors thank Doug Flewelling, Andrea Rodríguez, and Michael Worboys for helpful comments on the work while in progress.

9 References

Ackerman, E. (1994). Simulation of micropopulations in epidemiology: Tutorial 1. Simulation: an introduction, *International Journal of Bio-Medical Computing* 36: 229-238.

Al-Taha, K. and R. Barrera (1994). Identities through time. In: M. Ehlers and D. Steiner (eds.), *Proceedings of an International Workshop on Requirements for Integrated GIS*, New Orleans, LA, Environmental Research Institute of Michigan (ERIM), pp. 1-12.

Al-Taha, K. and R. Barrera (1990). Temporal data and GIS: An overview. *GIS/LIS '90*, Anaheim, CA, pp. 244-254.

Armstrong, M. (1988). Temporality in spatial databases. *GIS/LIS '88*, San Antonio, TX, pp. 880-889.

Bonfatti, F. and L. Pazzi (1995). Ontological foundations for state and identity within the object-oriented paradigm. *International Journal of Human-Computer Studies* 43: 891-906.

Catarci, T., G. Santucci, and M. Angelaccio (1993). Fundamental graphical primitives for visual query languages. *Information Systems* 18(2): 75-98.

Chen, P. (1976). The Entity-Relationship Model–toward a unified view of data. *ACM Transactions on Database Systems* 1(1): 9-36.

Chrisman, N. (1997). Beyond the snapshot: changing the approach to change, error, and process. In: M. Egenhofer and R. Golledge (eds.), *Spatial and Temporal Reasoning in Geographic Information Systems*. New York, NY, Oxford University Press, pp. 87-95. In Press.

Claramunt, C. and M. Thériault (1996). Toward semantics for modelling spatio-temporal processes within GIS. In: M. Kraak and M. Molenaar (eds.), *7th International Symposium on Spatial Data Handling*, Delft, The Netherlands, pp. 2.27-2.43.

Claramunt, C. and M. Thériault (1995). Managing time in GIS: an event-oriented approach. In: J. Clifford and A. Tuzhilin (eds.), *Recent Advances in Temporal Databases*. Berlin, Springer-Verlag, pp. 23-42.

Clarke, K., J. Brass, and P. Riggan (1994). A cellular automaton model of wildfire propagation and extinction. *Photogrammetric Engineering and Remote Sensing* 60(11): 1355-1367.

Cliff, A., P. Haggett, and D. Stroup (1992). The geographic structure of measles epidemics in the Northeastern United States. *American Journal of Epidemiology* 136(5): 592-602.

Cliff, A., P. Haggett, J. Ord, and G. Versey (1981). *Spatial Diffusion: An Historical Geography of Epidemics in an Island Community*, Cambridge, UK, Cambridge University Press.

DiBiase, D., A. MacEachren, J. Krygier, and C. Reeves (1992). Animation and the role of map design in scientific visualization. *Cartography and Geographic Information Systems* 19(4): 201-214.

Egenhofer, M. and J. Herring (1990). A mathematical framework for the definition of topological relationships. In: K. Brassel and H. Kishimoto (eds.), *Fourth International Symposium on Spatial Data Handling*, Zurich, Switzerland, International Geographic Union, pp. 803-813.

Egenhofer, M. and R. Golledge (1997). *Spatial and Temporal Reasoning in Geographic Information Systems*. New York, NY, Oxford University Press. In Press.

Frank, A. (1994). Qualitative temporal reasoning in GIS-ordered time scales. In: T. Waugh and R. Healey (eds.), *Sixth International Symposium on Spatial Data Handling,* Edinburgh, Scotland, pp. 410-431.

Frank, A. (1996). Qualitative spatial reasoning: cardinal directions as an example, *International Journal of Geographical Information Systems* 10(3): 269-290.

Helm, R., K. Marriott, and M. Odersky (1991). Building visual language parsers. In: (eds.), *CHI '91*, pp. 105-112.

Kainz, W., M. Egenhofer, and I. Greasley (1993). Modelling spatial relations and operations with partially ordered sets. *International Journal of Geographical Information Systems* 7(3): 215-229.

Khoshafian, S., and R. Abnous (1995). Object Orientation: Concepts, Analysis and Design, Languages, Databases, Graphical User Interfaces, Standards. New York, NY, John Wiley & Sons.

Khoshafian, S. and A. Baker (1996). *Multimedia and Imaging Databases.* San Francisco, CA, Morgan Kaufmann Publishers, Inc.

Khoshafian, S. and G. Copeland (1986). Object identity. *SIGPLAN Notices* 21: 406-416.

Kim, W., Ed. (1995). *Modern Database Systems: The Object Model, Interoperability, and Beyond.* New York, NY, ACM Press.

Kim, W., J. Banerjee, H-T. Chou, J. Garza, and D. Woelk (1987). Composite object support in an object-oriented database system, *OOPSLA'87 Proceedings, Special Issue of SIGPLAN Notices* 22(12): 118-125.

Langran, G. (1992). *Time in Geographic Information Systems*. Bristol, PA, Taylor & Francis Inc.

Laurini, R. and D. Thompson (1992). *Fundamentals of Spatial Information Systems*. London, UK, Academic Press.

Peuquet, D. (1994). It's about time: a conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers* 84(3): 441-461.

Raper, J. and D. Livingstone (1995). Development of a geomorphological spatial model using object-oriented design. *International Journal of Geographical Information Systems* 9(4): 359-383.

Schiel, U. (1989). Abstractions in semantic networks: axiom schemata for generalization. *SIGART Newsletter* 107: 25-26.

Shoham, Y. (1988). *Reasoning about Change*. Cambridge, MA, The MIT Press.

Sleezer, A. (1994). *Direct Manipulation of Temporally Constrained Activities for Geographic Modelling*. Master's Thesis, Department of Surveying Engineering. Orono, ME, University of Maine**.**

Smith, B. (1995). On drawing lines on a map. In: A. Frank and W. Kuhn (eds.), *COSIT '95*, Semmering, Austria, Springer-Verlag, Berlin, pp. 475-484.

Smith, B. (1996). *On the Origin of Objects*. Cambridge, MA, The MIT Press.

Smith, J. and D. Smith (1977). Database abstractions: aggregation. *Communications of the ACM* 20(6): 405-413.

Worboys, M. (1994). Object-oriented approaches to geo-referenced information. *International Journal of Geographical Information Systems* 8(4): 385-399.

Xu, J. and R. Lathrop (1995). Improving simulation accuracy of spread phenomenon in a raster-based geographic information system. *International Journal of Geographical Information Systems* 9(2): 153-168.

Yuan, M. (1994). Wildfire conceptual modeling for building GIS space-time models. *Proceedings of GIS/LIS '94*, Phoenix, AZ, pp. 860-869.