Three Compatible Mechanisms for Representing Medical Context Implicitly

Michel DOJAT[1], François PACHET[2]

[1]INSERM Unité 296, Faculté de médecine, 94010 Créteil, France.
[2]LAFORIA-IBP, Université Paris 6, Boîte 169, 4, Place Jussieu, 75252 Paris, France.
e-mails: dojat/pachet@laforia.ibp.fr

Abstract

An important aspect of context in medical reasoning is the notion of "variation" of a chunk of knowledge according to various contingencies, such as course of patient's disease, response to therapies, or team specificity. Our position is to represent these variations implicitly, by proposing mechanisms to factor knowledge and to refine it. We propose three mutually compatible mechanisms that effectively contribute to represent slight variations of knowledge in a representation framework integrating object-oriented programming and rule-based programming. We illustrate them with examples extracted from various knowledge bases for the management of mechanical ventilation.

Key words: Context, Knowledge Representation, Object-Oriented Programming, Rule-Based Programming.

1. Introduction

In medical reasoning, the notion of context is central for interpreting physiological parameters and is relevant at all levels of expertise. [Brézillon 93] showed that, within the community of researchers working on context issues, two main tendencies emerge:

<u>1) Context as a "situation"</u>

Context considered as something (ill-defined) designating the *situation* in which an interaction occurs, i.e. the place, the task, the participants, etc.
In the case of medical knowledge, context as situation encompasses several dimensions:

- context for the *interpretation of data*. Context reflects patient's history, type of disease or response to therapies.
- context to account for various *temporal phases* in the course of the patient's disease. Phases may correspond to various *tasks* occurring over time. In each phase, specific reasoning is envisaged.
- context to account for the *specificity of the clinical team*. Although clinical practice is largely based on consensual medical knowledge, each clinical team, specialized in a particular medical field, accumulates specific experience. This difference with commonly accepted knowledge reflects habits, cultural contexts, specific skills and experiences.

2) Context as a "set of beliefs"

Context considered as a *restricted set of beliefs* or logical assumptions that are necessary to interpret correctly available information. In the medical field, it is often important to explicitly represent the underlying assumptions of the medical doctor regarding the patient's status, such as the diagnosis established, the theoretically predicted evolution of the patient's state or the expected evolution in function of the current treatment.

In medical expertise, knowledge to be used in several contexts bear a lot of similarities. As we will see in the examples, a vital aspect of context is the notion of *variation* of a chunk of knowledge according to various contingencies, such as patient's response to therapy, course of patient's disease or team specificity. Representing similarities between different contexts raises complex issues. For instance, [Guha 91] proposes a dedicated mechanism (the so-called *lifting rules*) to establish some kind of communication between different but related contexts.

Moreover, context is *multiform* and the mechanisms proposed to account for it in the representation of knowledge, are various and often difficult to conciliate. For example, the context mechanism proposed by [Guha 91] is hard to transpose in a non-logical representation framework. Similarly, the mechanism proposed by [Abu-Hakima 93] requires the explicit representation of goals to be used by the inference process.

Based on these two remarks, our position regarding context is 1) to focus on *mechanisms*, rather than on explicit representations, and 2) focus on *similarities* rather than on dissimilarities, and 3) ensure some kind of *compatibility* of various manifestations of context-dependency. We propose three *mutually compatible* mechanisms for factoring medical knowledge, thereby providing a powerful implicit representation of context. These three mechanisms are: *class inheritance*, *natural typing* and *rule base inheritance*. They are applicable in representation frameworks integrating rule-based programming with object-oriented programming.

Our research is being conducted in collaboration with a team specialized in intensive care medicine at Henri Mondor Hospital (Créteil, France). We developed a knowledge-based system for ventilator management, called NéoGanesh. NéoGanesh adjusts *in closed-loop* the mechanical assistance, depending on the patient's physiological needs, and indicates when the patient can breathe without external assistance (*weaning*). The results [Dojat et al. 92], [Dojat et al. 94] demonstrated that NéoGanesh improves the quality of the patient's ventilation, and therefore validates our approach. To illustrate the interest of our proposals, we will use examples extracted from NéoGanesh and from expertise in the same area found in the literature.

2. Our representation framework: Alliance of Objects and Rules

Object-oriented languages propose to represent knowledge by *simulation*. Software objets are constructed to represent a monitor, a patient, or a clinician, as well as therapies, diagnosis, expectations and so forth. Operations on these objects represent domain operations such as monitoring respiratory rate, changing respiratory rate and administrating a new therapy, and are represented as *methods* associated to corresponding classes.

The need for combining object structures and rule-based programming has been widely recognized. The fact base of a rule-based program is a model of the concrete situation that is currently being processed. To bring some semantic structure to facts, one naturally tends to see them as properties of objects that build up a universe simulating the concrete world. Individual facts are no longer represented as such, their logical value is ascertained by querying objects in the model. The fact base is thus dissolved into an object-oriented model of the world. The equation "fact base = model of the world" links the two techniques.

Starting with a standard object-oriented language (Smalltalk-80), and extending it with an embedded rule-based layer, we built the NéOpus system [Pachet 95]. Turning to classical object-oriented style (the so-called message passing) causes some trouble to the knowledge representation specialist (see, e.g. [Nebel 90]). We refer to [Pachet 94] for a discussion on the technical problems posed by this integration. However, we feel that the benefit gained from potentially applying rules to the whole universe of object-oriented models created by object-oriented programmers does warrant these discrepancies from the generally accepted principles.

3. Three Mutually Compatible Mechanisms to Account for Context

We propose three mutually compatible mechanisms that address three different *levels* of granularity for factoring medical knowledge representing context in our object-oriented setting 1) class inheritance, 2) natural typing, and 3) rule base inheritance.

3.1. Class Inheritance to Account for Context

The organizing principle of *class inheritance* allows to describe medical information and create information structures comprising concepts which are statically related through common property characteristics. Although class inheritance can hardly be compared with true classification mechanisms as found, e.g. in description logics [Patel-Schneider 90], it nevertheless allows us to represent taxonomies of physical or conceptual entities and generalization/specialization relation (e.g. class `Intensivist` inherits from class `Clinician` which in turn inherits from `MedicalDoctor`). Here is an example that shows how class inheritance can be used effectively to account for context.

Availability of data as Context

Well-structured problems of diagnosis are solved by the method of *heuristic classification*. This method requires three steps: 1) abstraction from input data, 2) association with a taxonomy of possible solutions and 3) refinement of the collection of potential solutions [Clancey 85]. Embedding classification mechanisms in object-oriented languages is not, however a trivial task [Yelland 92]. For our applications, we propose to use class inheritance to represent the taxonomy of possible solutions. Each class holds a set of constraints to which patient's data is matched for classification. For instance, to diagnose the current respiratory state of a patient, clinicians currently use three main parameters: respiratory rate (RR), volume inspired at each breathing cycle (Vt), and pressure of carbon dioxide at the end of the expiration ($PetCO_2$). Consequently, the class `Normal` representing a normal ventilation, holds the following set of constraints (expressed as a single Boolean expression):

```
Constraint for class "Tachypnea"
        VT > 250 AND
        (RR < 35 AND RR > 28) AND
        PetCO2 < 55
```

The classification process consists in matching a given patient ventilation state (represented by an instance of class `MeasuredVentilation`) against the solution classes. For instance, the solution classes that require parameters VT, RR, and $PetCO_2$ are: `Normal`, `Tachypnea`, `Bradypnea`, `Insufficient`, and so forth. The definition attached to each element of the taxonomy of solutions, as well as the thresholds used for interpreting physiological data, may vary depending to the context: the patient's characteristics (sex, age, pathology, etc.), the clinical evolution, or the medical doctor. Moreover, the refinement of the potential solutions may depend on the *availability* of certain data. For instance, to characterize the patient's respiratory state more precisely, we can *add* a parameter, called *Occlusion Pressure* or P0.1, which measure the patient's effort needed to trigger the mechanical assistance. To represent this new piece of information (and hence a new context of interpretation of data), we add new solution classes to the taxonomy, with constraints that explicitly refer to this new data. We add the solution class `FineTachypnea` to our taxonomy by defining a *subclass* of `Tachypnea`. Similarly, we will subclass `Normal` with class `FineNormal`, holding a constraint with an additional condition on P0.1, and so forth.

Instead of systematically matching against all the classes of the taxonomy, we organize solution classes in several collections, each one representing a particular combination of available parameters. These collections of classes are represented as methods in the class that represents the current clinician (`Intensivist`). `standardSolutionsList` represents the list of solution classes when RR, Vt and $PetCO_2$ are available, `fineSolutionsList` is used if P0.1 is available and so forth.

3.2. Natural Typing to Account for Context

The preceding example shows how inheritance may be used to factor information. *Encapsulation* (or information hiding) supplies a useful barrier among several levels of abstraction. Now class inheritance and encapsulation may be used *in conjunction with* rules, yielding another dimension of context representation. This is realized in NéOpus by so-called *natural typing* of rule variables.

3.2.1. The Mechanism

The idea behind natural typing is to allow the pattern-matcher to consider direct instances as well as instances of subclasses to be matched by rule variables for a given rule. The interpretation of a rule is therefore *dynamic*, since 1) the condition and action parts of the rules are entirely expressed in terms of messages sent to the matched objects and 2) the messages are redefined in subclasses. In other words, the rules are *context-dependent*, where the context is represented by the set of objects that match the rule.

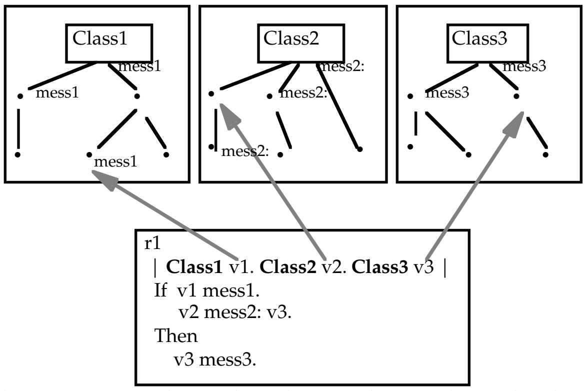


Figure 1. A rule with the corresponding class hierarchies. The rule should be read as follows: For any `v1`, instance of class `Class1`, `v2` instance of `Class2` and `v3` instance of `Class3`, if "`v1 mess1`" and "`v2 mess: v3`" evaluate to true, then evaluate "`v3 mess3`". The rule uses three messages (`mess1` through `m3`) which are defined and redefined in the classes of the corresponding hierarchies.

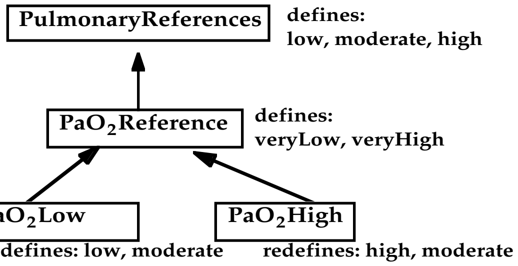
More precisely, let `r` be a rule that declares n variables v_i (i=1, n). Let C_i be the class declared for v_i. Now, if each C_i has k_i concrete subclasses (Cf. Figure 1), the total number of possibly different interpretations of `r` is $\Pi\ k_i$ (i = 1, n). In practice, if r has 3 variables, and each declared class has 5 concrete subclasses, then r has $5^3 = 125$ possibly different interpretations !

3.2.2. Reference Values as Context

[East et al. 94] develop computerized protocols to assist clinicians with the difficult task of ventilating mechanically patients with *Acute Respiratory Distress Syndrome* (ARDS). They identify the arterial oxygenation of the patient by means of PaO_2 measurement to define the best therapy to provide. If the clinician diagnoses a Barotrauma, he tolerates a lower arterial pressure of oxygen, and tries to reduce alveolar pressures. We represent *references of PaO_2* by classes (Cf. Figure 2, left), and their variations by subclasses. In this scheme, the reference used for PaO_2 in the case of Barotrauma is therefore represented by a class `PaO2Low`, subclass of `PaO2Reference`, which redefines method "`low`" with the appropriate value.

The chosen strategy, function of the diagnosis performed, determines the appropriate subclass of `PaO2Reference` that should be used for classifying PaO_2. The actual arterial oxygenation of the patient is itself represented by an instance of `MeasuredVentilation`. The rule (Cf. Figure 2, right), is then used to classify arterial oxygenation by comparing it with the chosen PaO2 reference.

Thanks to natural typing, the condition part of the rule is interpreted differently depending on the class of the reference of PaO_2.




```
acceptableOxygenation
|MeasuredVentilation   vent.
PaO2Reference reference|

vent paO2 > reference low.
vent paO2 < reference moderate.

then
Oxygenation new acceptable.
```

Figure 2: Left: gradual introduction of modification of thresholds for arterial pressure of oxygen classification. Right: an example of rule to classify arterial oxygenation. Methods `low` and `moderate`, which return corresponding threshold values, are redefined in subclasses of `PaO2Reference`.

3.2.3. Management of Time

In most medical knowledge-based systems, the course of the patient's disease is divided into several temporal phases. These phases may correspond to *steps* in the treatment, e.g. in cholesterol treatment [Rucker et al. 90], or respiratory assistance [Fagan 80], or various *states* of the patient's evolution, such as hemodynamic states ([Lau and Vincent 93], [Cohn et al. 90]). In each phase, specific rules are applied.

Recently, [Dojat and Sayettat 94] proposed a model for temporal reasoning representation in real-time systems. This model is based on temporal abstractions that allow observations to be interpreted incrementally as they are acquired. Two mechanisms are used: *aggregation* of similar situations and *forgetting* non-relevant, redundant or out of date information. Activation of these two mechanisms are context-dependent.

We implement this model using our combination of class inheritance and rules. This is performed by 1) reifying phases, and building a hierarchy of classes representing various significant phases of the course of the patient's disease and 2) introducing an explicit management of time that account for state change and evolution.

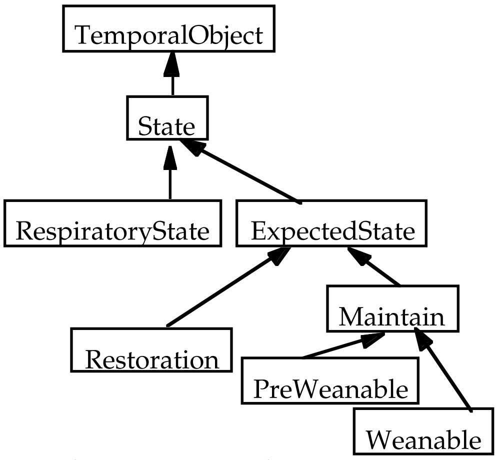


Figure 3: Excerpt of the hierarchy of temporal objects in NéoGanesh.

Temporal objects are time-stamped entities, used to develop a temporal discourse about the patient's ventilation evolution. The figure 3 shows parts of the hierarchy of temporal objects used in NéoGanesh. `RespiratoryState` represents the ventilation status of the patient. `ExpectedState` represents the future patient's respiratory state, as expected by the doctor. One of the goal of the medical reasoning is to appreciate the stability of the ventilation over periods of time, the duration of instabilities and the persistence of inadequate therapies. Thus, the clinician has to perceive significant changes between successive observed states or successive expected states. This perception of changes is, of course, context-dependent: for instance, the notions of *similarity* and *dissimilarity* between states depends on the class of the states compared. This is, once again, represented by methods attached to the corresponding subclasses of `TemporalObject`. Figure 4 shows a rule taken from the rule base that handles perception of change.

```
partialContinuity
|State s1 s2 s3. Duration d1|
 s3 persistent.
 s1 similarTo: s3.
 s2 dissimilarTo: s3.
 s2 between: s1 and: s3.
 s2 duration <= d1.
then
     s1 validDuring: s3 duration.
```

Figure 4: A rule that aggregates two disjoint states (`s1` and `s3`), separated by an instability (`s2`) considered short enough to be discarded.

NéoGanesh includes numerous other examples of the use of natural typing, including more sophisticated management of thresholds and representation of control ([Dojat and Pachet 92]).

3.3. Rule Base Inheritance to Account for Context

We introduce a third level of context-dependency: rule base inheritance (RBI). RBI allows to specify gradually the context of application of rules by matching objects that are relevant for this context.

3.3.1. The Mechanism

In our representation framework (NéOpus), rules are grouped in rule bases that are represented as abstract classes. The idea of transposing the class inheritance mechanism to rule bases is therefore natural. We introduced in [Pachet and Perrot 94] a scheme to transpose the *intuition* of class inheritance to rule bases. In this scheme, each rule base may be defined as a sub-base of an existing rule base, thereby inheriting all its rules. An overriding mechanism, based on rule names, allows a rule base to *redefine* a inherited rule into a more specific rule. This scheme is based on a static propagation of rule compilation to sub-bases, together with a particular *control strategy* (RBI strategy) that ensures that, in case of conflict, rules defined in the lowest sub-base will be selected.

There are indeed several advantages in providing rule bases with an inheritance mechanism. It provides a high level scheme for organizing rules, it allows to factor out common rules, and simplifies control strategy specification.

We will show here how RBI may be used to provide explicit contextual dependency of medical knowledge.

3.3.2. RBI for Context Dependency in Temporal Reasoning

The Figure 5 shows the inheritance tree for rule bases, which gradually introduce context-dependency in temporal management, at the rule base (or task) level.

Rule base `TemporalManagement` contains rules that define the perception of continuity, discontinuity, partial continuity and so forth, in a use-neutral manner. They match general temporal objects (`State`).

Rule base `VentilationEvolution` introduces a refinement of this general-purpose temporal reasoning to adapt it to the ventilation management. `VentilationEvolution` inherits all the rules defined in its super-base, redefines some of them (such as `partialContinuity`), and defines specific rules (such as `extendedAggregation`). Rules defined or redefined in `VentilationEvolution` talk about `RespiratoryState` (instead of the general `State` class, see Figure 6).

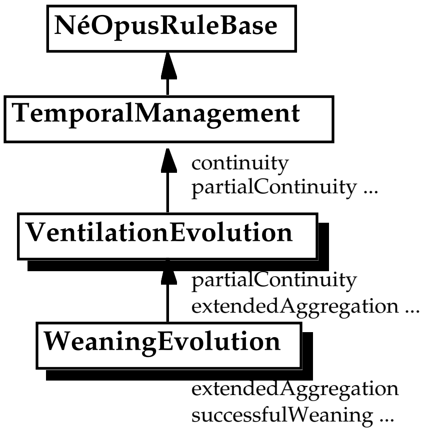


Figure 5: An inheritance tree of rule bases.

```
partialContinuity (redefined in VentilationEvolution)
   |RespiratoryState s1 s2 s3|
 s3 persistent.
 s1 similarTo: s3.
 s2 dissimilarTo: s3.
 s2 between: s1 and: s3.
 s2 durationInExpertise <= 1.
 s3 durationInExpertise > 1.
then
   s1 validDuring: s3 duration.
```

Figure 6: Rule `partialContinuity` redefined in a sub-base of `TemporalManagement`. The rule matches more specialized objects (`RespiratoryState` instead of `State`), modifies and adds a condition. The rule states that 2 disjoint respiratory states, separated by an instability that lasts less than 1 expertise cycle, can be aggregated.

Finally, `WeaningEvolution` is concerned by situations when the patient is on the verge of breathing without assistance. `ExtendedAggregation` rule is modified to tolerate moderate ventilation instabilities. `SuccessfulWeaning` indicates to the user that the patient can be extubated. At this level, rules match only instances of class `Weanable`.

4. Discussion

We have introduced three mutually compatible mechanisms that take context into account. We illustrated the mechanisms by instanciation of context-dependency for *availability of data* (class inheritance), representing *reference values* (natural typing in forward-chaining rules), and representing different *perception to changes* (rule base inheritance). Object-orientation casts a new light on the distinction between the two notions of context (situation versus set of beliefs). Because our representation model is uniform (everything is represented by an object), the notion of a "logical fact" disappears, making it difficult to represent "sets of beliefs". However, we think that this distinction is still relevant, and propose to take it into account by distinguish between two categories of domain objects:

- *perceived* objects (symptoms, devices, patient's status and history), that represent the world as it is "perceived" by the clinician expert. These objects represent concrete real-world objects as well as objects pre-existing to the reasoning process.
- *conceived* objects (diagnosis, models of ventilation and expectations) that represent the world as it is "conceived" by the clinician expert, and created or worked out by his reasoning process.

In this setting, the distinction situation/belief is revisited: situation is represented by perceived objects, and beliefs by conceived objects.

Few systems propose precise descriptions of the implementation level and how they practically achieve some forms of context representationIn this paper we identified three mechanisms of rule-based object-oriented programming: class inheritance, natural typing and rule base inheritance and showed how they allow to implement various dimensions of context-dependency in medical knowledge bases. This reinforces the potential interest of the object-oriented paradigm and production rules association. We are convinced that the development of large open health care systems might strongly benefit from this approach.

5. References

[Abu-Hakima 93] **Abu-Hakima S.**, *The Use of Context in Diagnostic Systems*, in Proceedings IJCAI'93 Workhop on Using Knowledge In its Context, Chambéry, France, Université Paris 6, pp. 13-19, 1993.

[Brézillon 93] **Brézillon P.**, *Proceedings of the IJCAI 93 Workshop on Using Knowledge in its Context*, Technical Report 93/13, LAFORIA-IBP, Université Paris 6, 1993.

[Clancey 85] **Clancey W.J.**, *Heuristic classification*, Artificial Intelligence, vol. 27, pp. 289-350, 1985.

[Cohn et al. 90] **Cohn A.I., Rosenbaum S., Factor M. and Miller P.L.**, *DYNASCENE: An approach to computer-based intelligent cardiovascular monitoring*

using sequential clinical "scenes", Methods of Information in Medicine, vol. 29, pp. 122-131, 1990.
[Dojat et al. 92] **Dojat M., Brochard L., Lemaire F. and Harf A.**, *A knowledge-based system for assisted ventilation of patients in intensive care*, International Journal of Clinical Monitoring and Computing, vol. 9, pp. 239-250, 1992.
[Dojat et al. 94] **Dojat M., Harf A., Touchard D., Laforest M., Lemaire F. and Brochard L.**, *Clinical evaluation of a knowledge-based system providing ventilatory management and decision for extubation during weaning from mechanical ventilation*, American Review of Respiratory Disease, vol. (submitted), 1994.
[Dojat and Pachet 92] **Dojat M. and Pachet F.**, *NéoGanesh: an extendable knowledge-based system for the control of mechanical ventilation*, in Proceedings 14th annual international conference of IEEE-EMBS, Paris, pp. 920-921, 1992.
[Dojat and Sayettat 94] **Dojat M. and Sayettat C.**, *A realistic model for temporal reasoning in real-time patient monitoring*, Applied Artificial Intelligence, (to appear), 1994.
[East et al. 94] **East T., Morris A.H., Wallace C. J., Clemmer T. P., Weaver L. K. and Orme J. F.**, *Computerized protocols for the management of arterial oxygenation in patients with ARDS on assist control mechanical ventilation*, American Review of Respiratory Disease, (to appear), 1994.
[Fagan 80] **Fagan L.M.**, *Representing time dependant relations in a medical settings*, Ph.D. thesis, University of Stanford (Ca), 1980.
[Guha 91] **Guha R.V.**, *Contexts: A formalization and some applications,* Technical Report ACT-CYC-423-91, MCC, 1991.
[Lau and Vincent 1993] **Lau F. and Vincent D. D.**, *Formalized decision support for cardiovascular intensive care*, Computers and Biomedical Research, vol. 26, pp. 294-309, 1993.
[Nebel 90] **Nebel B.**, *Reasoning and Revision in Hybrid Representation Systems*. Lecture Notes in Artificial Intelligence, Berlin: Springer-Verlag, 1990.
[Pachet 94] **Pachet F.**, *Report on the OOPSLA'94 Workshop on EOOPS*, in Proceedings Addendum to the OOPSLA'94 proceedings, Portland, Oregon, ACM Sigplan notice, 1994.
[Pachet 95] **Pachet F**, *On the embeddability of production systems in object-oriented languages*, Journal of Object-Oriented Programming, (to appear), 1995.
[Pachet and Perrot 94] **Pachet F. and Perrot J.-F.**, *Rule Firing with Metarules*, in Proceedings SEKE'94, Jurmala, Latvia, Knowledge Systems Institute, pp. 322-329, 1994.
[Patel-Schneider 90] **Patel-Schneider P. F.**, *Practical, Object-Based Knowledge Representation for Knowledge-Based Systems*, Information Systems, vol. 15, pp. 9-19, 1990.
[Rucker et al. 90] **Rucker D.W., Maron D.J. and Shortliffe E.H.**, *Temporal representation of clinical algorithms using expert system and database tools*, Computers and biomedical research, vol. 23, pp. 222-239, 1990.
[Yelland 92] **Yelland Ph. M.**, *Experimental Classification Facilities for Smalltalk*, in Proceedings OOPSLA '92, Vancouver, Canada, pp. 235-246, 1992.