GRPR-RT-9608
February 1996

Gnome: A GeNeric Object ModEl

Frédéric Labrosse **Jean-Yves Hervé** **Paul Cohen**

Groupe de Recherche en Perception et Robotique

École Polytechnique de Montréal
GRPR - DGEGI
P.O. Box 6079, Station Centre-ville
Montréal (Québec) CANADA H3C 3A7

[labrosse,jyh,cohen]@ai.polymtl.ca

Abstract

This document presents a GeNeric Object ModEl (GNOME), which was previously introduced in [Lab95]. Gnome allows the representation of objects through their various attributes (geometry, color, density, *etc.*) in a hierarchical and topological manner on the one hand and quantitatively on the other hand: each attribute is represented by a tree of attribute-objects (objects defined in the attribute's representation space) and by a graph representing relations between these attribute-objects. Nodes and arcs quantifications allow the specification of dimensions, forms, values, *etc.* Finally, relations between attributes, separately represented until now, are introduced. This report concludes with a comparison of GNOME with different models in the literature. This report is also available in French under the same technical report number.

This work was partially supported by the NSERC-Noranda chair in mining automation.

Published in February 1996.

GNOME: A GeNeric Object ModEl

GRPR-RT-9608

Frédéric Labrosse, Jean-Yves Hervé, Paul Cohen
(Groupe de Recherche en Perception et Robotique)

To get a copy of this document, see the following electronic address:

`http://www.ai.polymtl.ca/`

or contact:

Groupe de Recherche en Perception et Robotique
École Polytechnique de Montréal
Pavillon André Aisenstadt — room 5467
P.O. Box 6079, Station Centre-ville
Montréal (Québec) CANADA H3C 3A7

Phone: (514) 340-5783
Fax: (514) 340-5802

Contents

List of Figures

1 Introduction

One can have many different representations of a given object, depending on the task involving the object or the agent performing the task. In the field of robotics, tasks and agents affecting an object can take various forms and, moreover, may sometimes change, thereby implying a change of representation. This observation motivates the development of task- and agent-independent representation models. That is why we propose in this report a GeNeric Object ModEl (GNOME).

In GNOME, the objects are represented by their various attributes, which can be, *e.g.*, geometry, color, density, or temperature. For each attribute, an arborescent structure is created by the hierarchical decomposition of *attribute-objects* (objects defined in the representation space of the considered attribute). A graph representing relations between attribute-objects is grafted on this tree. These relations can be connection, relative pose, order, *etc.* Each attribute is thus represented, independently of the others, by a tree and a graph constituting a topological part of the model (the *structure*). The association of applications (in the mathematical meaning) to attribute-objects and relations between these attribute-objects creates a quantitative part (the *quantification*) that allows the specification of the represented object's size, form, *etc.* Last, another graph is introduced to represent *inter-attribute* relations.

Road Network (1): All along the report, the concepts described will be introduced by the example of a representation, among all the possible ones, of a road network with its car traffic, speed limits, pavement state, *etc.* The network is supposed to be a non-urban one. All the document's parts called "Road Network" are part of this example.

Representation (1.a): In the document's parts entitled "Representation," the developed formalism is applied to the corresponding example part.

Section 2 gives some notations required for the model's description, which is presented in Section 3. More precisely, the structure, the quantification, and the inter-attributes relations are respectively presented in Sections 3.1, 3.2, and 3.3. A second example, essentially graphical, is given in Section 4. Finally, a comparison between GNOME et several models in the literature is proposed in Section 5. A list of symbols is given at the end of the document.

2 Notations

A *graph* $\Gamma(A)$ is a set of pairs (the arcs) whose elements belong to A (Figure 1(a)):

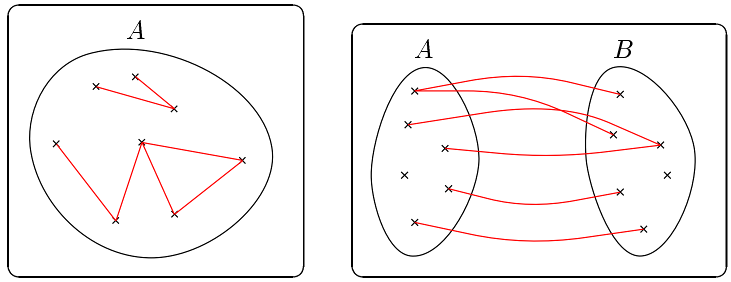
$$\Gamma(A) \subset \{(a_1, a_2) \in A \times A\}.$$

If the graph represents a particular relation, it is noted $\Gamma_r(A)$ and:

$$\Gamma_r(A) = \{(a_1, a_2) \in A \times A \mid r(a_1, a_2)\},$$

where $r(a_1, a_2)$ means that a_1 is in relation with a_2 under relation r. A *bipartite graph*, $\Gamma(A, B)$, is a graph whose pairs are elements of $A \times B$ (Figure 1(b)):

$$\Gamma(A, B) \subset \{(a, b) \in A \times B\}.$$



(a) In a set (b) Between two sets

Figure 1: Graphs over one or two sets

The graphs introduced until now are binary ones, *i.e.* the arcs are pairs. A *n-ary graph* is a graph whose arcs are n-tuples. This allows the introduction of a *n-partite graph*, $\Gamma(A_1, \cdots, A_n)$, which is a graph whose arcs are n-tuples of $A_1 \times A_2 \times \cdots \times A_n$.

Whenever a graph Γ represents several relations, it is transformed into a *labeled graph*. In order to do so, a function F is associated with it (the definition of a function is given later). This function has the set of arcs as domain and associates labels representing the relations (and whose set is E) to each arc. Thus:

$$F\colon \Gamma \to \mathcal{P}(E),$$

where $\mathcal{P}(E)$ is the set of parts of E[1].

A path $\Pi(\Gamma)$ in the graph Γ is a set of arcs of Γ constituting a succession of Γ's nodes:

$$\Pi(\Gamma) = \{(a_i, a_{i+1})\ i = 0, \cdots, n \mid (a_i, a_{i+1}) \in \Gamma\}.$$

A tree $\Theta(A)$ is a graph defined over the set A with the following properties:

- it is connected, *i.e.* there exists a path between any two elements a_1 and a_2 of A, and
- there is no cycle in the graph, *i.e.* there is no path from an element a of A to itself.

The first [resp. second] projection of a graph $\Gamma(A, B)$ is the set $\mathrm{pr}_1\Gamma$ [resp. $\mathrm{pr}_2\Gamma$] such that:

$$\forall a \in A, (a \in \mathrm{pr}_1\Gamma) \Leftrightarrow (\exists b \in B \mid (a, b) \in \Gamma)$$
$$[\text{resp. } \forall b \in B, (b \in \mathrm{pr}_2\Gamma) \Leftrightarrow (\exists a \in A \mid (a, b) \in \Gamma)].$$

A *correspondence* from A to B is a triplet $\mathrm{C} = (\Gamma, A, B)$, where A and B are two sets and Γ is a graph from A to B. $\mathrm{pr}_1\Gamma$ is the domain set and $\mathrm{pr}_2\Gamma$ is the range. By extension (and to facilitate the notations), $\mathrm{pr}_1\mathrm{C}$ [resp. $\mathrm{pr}_2\mathrm{C}$] will be the set $\mathrm{pr}_1\Gamma$ [resp. $\mathrm{pr}_2\Gamma$].

A functional graph is a graph $\Gamma(A, B)$ such that each element of A has at most an image in B. A *function* is a correspondence whose graph is functional.

[1] [Big85] defines a *weighted* graph as a graph with a cost function from the set of arcs to the set $\mathbb{N}$.

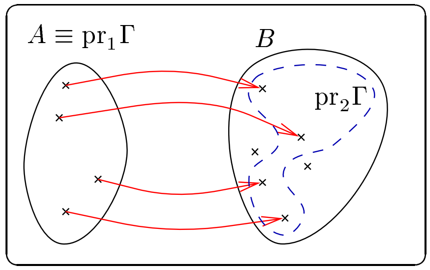


Figure 2: An application

An application $f = (\Gamma, A, B)$ is a correspondence for which each element of A, the domain, has one and only one element in B (Figure 2):

$$\forall a \in A, \exists! b \in B \mid (a, b) \in \Gamma.$$

The set of all the applications from A to B is $\mathcal{F}(A, B)$. When the sets are not specified, $\mathcal{F}$ is the set of all the applications.

3 The Model

Road Network (2): Our road network has a geometry: the roads' length, curvature, and width, the roads' relative pose, *etc.* The network is also characterized by non-geometric attributes, for example, the car traffic on every network point and the pavement state.

In GNOME, the objects are represented through their attributes, which can be their geometry, color, temperature, density, *etc.* The k-th attribute is noted A_k. By definition, A_0 is the geometric attribute and there is N non-geometric attributes.

Road Network (3): The car traffic depends on the density of cars, but also on their speed. Moreover, at a given network point, the traffic varies with the time of the day. On the other hand, on a given road, the traffic can have one or two directions, depending on whether the road is one way or not. A road, at least in North America, can be oriented "North-South" or "East-West." This gives four possibilities for the car traffic (traffic toward the North, the South, the East, or the West). A fifth possibility is to be added: "no traffic," to indicate that the road is one-way(2). The traffic is thus represented, in our example, in a four-dimensional space: one for the density, one for the speed, one for the time, and one for the direction.

At each network point there is a traffic value described by four parameters. This gives a cloud of points in a 4-D space(3). This cloud can be seen from different distances, which lets appear point groupings corresponding to different levels.

(2)This fact could be represented by a nul density, for example. But given that the traffic varies with time, it can be nul (nul density) for periods of time. In this case, it is impossible to distinguish a nul traffic at a given time from a traffic that can't exist because of a one-way road. This is an important information to to plan a road-map, and thus should be represented in the model.

(3)We can note that the traffic's representation does not contain the road network points that hold the traffic values. There are some correspondences which will make the connection between the traffic representation space and the network geometry (see Road Network (13)).

Each attribute A_k will be represented in a particular space, *i.e.* the attribute takes its values, as points, in this space. These points, grouped following different criteria corresponding to different levels, form clusters which we will call *attribute-objects*. So, for each attribute, there is a whole set of possible attribute-objects. From now on, the symbol **A** will be used to indicate a generic attribute-object. The indices are omitted when we are not interested in a particular attribute.

Road Network (4): Let us examine the road network geometry. At the first representation level, the network is split into a set of roads and intersections, a road being delimited by two intersections. Next, still in our particular representation, each road is split into road sections following a given criterion: for example, the speed limit.

Some relations will be represented between roads and intersections on one hand and between road sections on the other. The first relation is the connection which indicates which road is connected to which intersection and which road section is connected to which road section. Four other relations are "is North of," "is South of," "is East of," and "is West of."

A value is associated to each road, its length, and to each intersection, its position in a terrestrial system of reference. The road sections are represented as ribbons: a 3-D parametric curve in the terrestrial system of reference and a width for each curve point(4).

To each road corresponds a traffic value: a mean value of density and speed over a day for each road direction(5). To each point of the road section 3-D curve (the one in the terrestrial system of reference) corresponds two car traffic values(6). These values are functions of time.

The representation $\mathcal{R}$ of an object **O** is a triplet made of a *structure*, a *quantification*, and a graph representing the *inter-attributes relations*. Notice that object **O** is not an attribute-object of the geometric attribute but The Object which is being represented with all its attributes.

We will now study each element of the triplet in the next sections.

3.1 The Structure

Each attribute is represented by a tree of hierarchical decomposition of attribute-objects and a graph representing the relations that exists between the attribute-objects that constitute the nodes of the tree. The structure of object **O** is the set of all the attributes' trees and graphs.

3.1.1 The Tree of Attribute-objects Decomposition

Road Network (5): Road Network (4) tells us that the road network is split into roads and intersections. Each road is itself split into road sections. This decomposition relation is represented by a tree.

---

(4)This gives a 4-D curve.

(5)If the road is a one-way one, one traffic value is nul.

(6)Despite the fact that there is no intersection on the roads (They have been "removed" at the previous stage), the traffic is not constant as is the car flow.

Given $\mathbf{A}$, an attribute-object from the A_k attribute, *i.e.* belonging to the set of all the attribute-objects definable in the attribute's representation space. This attribute-object is split into attribute-subobjects of this same set. These attribute-subobjects are thus $\mathbf{A}$'s descendants. Their set is $\Delta_k(\mathbf{A})$ and a generic attribute-subobject will be noted $\mathbf{a}$. Let us note that an attribute-object may have many, one, or no attribute-subobject.

Each attribute-object of $\Delta_k(\mathbf{A})$ can also be split in attribute-subobjects to obtain finer representation level. This decomposition relation is recursive and, thus, can be represented as an *attribute-objects decomposition tree* $\Theta(\Omega_k)$ whose nodes are the attribute-objects and whose arcs show the decomposition. The set of all the tree's attribute-objects, or nodes, is Ω_k.

If some attribute-objects are topologically (in the meaning of their form, as opposed to their size, for example) identical, then they are represented by a same attribute-object in Ω_k (see Figure 6).

More generally, each arc of the decomposition relation is quantified by an application which gives the number of instances of each attribute-object of Ω_k involved in the decomposition.

The attribute-object tree $\Theta(\Omega_k)$ is thus labeled with the following function:

$$F\colon \Theta(\Omega_k) \to \mathbb{N}. \tag{1}$$

Figure 3 shows an attribute-objects decomposition tree. For clarity reasons, the arc labels are not shown (the continuous line arcs are explained in Section 3.1.2).

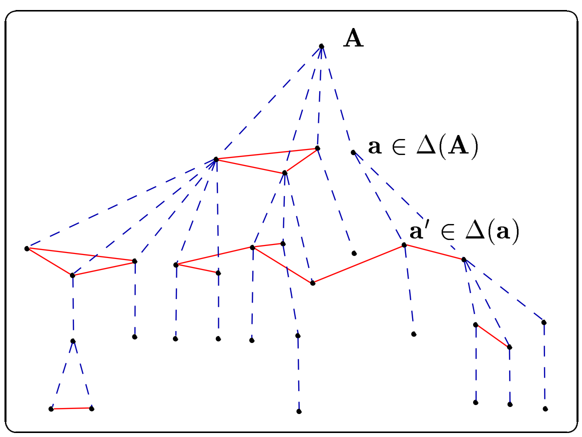


Figure 3: A decomposition tree and a relation graph. The dashed lines show the attribute-object hierarchical decomposition while the continuous lines show the relations between attribute-objects (*e.g.* connection).

Representation (5.a): Let us formalize the road network decomposition tree for the geometric attribute. The descendants of attribute-objects *road network* (R) are *roads* (r_k) and *intersections* (i_l):

$$\Delta_0(\mathrm{R}) = \{\mathrm{i}_l\} \cup \{\mathrm{r}_k\},$$

where $\{\mathrm{i}_l\}$ is the set of all intersections and $\{\mathrm{r}_k\}$ is the set of all roads. We define the descendants of a road from $\Delta_0(\mathrm{R})$ as a set of *sections* (t_k):

$$\Delta_0(\mathrm{r}) = \{\mathrm{t}_k\}.$$

The attribute-object set is:

$$\Omega_0 = \Delta_0(\mathrm{R}) \bigcup_k \Delta_0(\mathrm{r}_k).$$

3.1.2 Relations Between Attribute-objects of Ω_k

Road Network (6): In the proposed representation of the road network, we choose to incorporate connections and cardinal relations (see Example 4). These relations can be represented by a graph, the nodes of which are roads, intersections, and road sections, and whose arcs indicate the attribute-objects in relation.

There exists relations between the attribute-objects of Ω_k, which can be connection between attribute-objects, relative pose, *etc.* All relations are represented by a single graph $\Gamma(\Omega_k)$, whose nodes are attribute-objects of Ω_k, and whose arcs indicate the attribute-objects in relation. Since there are several relations, the graph's arcs are labeled to indicate which relations they represent. If R is the set of labels representing the relations, a function F_k is associated with graph $\Gamma(\Omega_k)$:

$$\mathrm{F}_k\colon \Gamma(\Omega_k) \to \mathcal{P}(R).$$

As was shown in Section 3.1.1, the decomposition tree nodes can represent several attribute-objects, as in the example of Section 4. Thus, the arcs have to include a supplementary label (an element from $\mathbb{N}$) which permits the distinction between the different attribute-object instances. The previous equation become:

$$\mathrm{F}_k\colon \Gamma(\Omega) \to \mathcal{P}(R \times \mathbb{N}). \tag{2}$$

Representation (6.a): The graph representing the relations between the geometric attribute-object is splittable into parts which correspond to particular relations:

$$\Gamma(\Omega_0) = \Gamma_c(\Omega_0) \cup \Gamma_s(\Omega_0) \cup \Gamma_n(\Omega_0) \cup \Gamma_e(\Omega_0) \cup \Gamma_o(\Omega_0),$$

where Γ_c is the graph representing the connection, Γ_s the one representing the relation "is South of," *etc.* The subgraph representing connections can thus be written as:

$$\begin{aligned}\Gamma_c(\Omega_0) = {} & \{(\mathrm{r},\mathrm{i}) \in \Delta_0(\mathrm{R}) \times \Delta_0(\mathrm{R}) \mid \mathrm{r} \text{ is connected to } \mathrm{i}\} \cup \\ & \{(\mathrm{t}_1,\mathrm{t}_2) \in \Delta_0(\mathrm{r}) \times \Delta_0(\mathrm{r}) \mid \mathrm{r} \in \Delta_0(\mathrm{R}), \mathrm{t}_1 \text{ is connected to } \mathrm{t}_2\}.\end{aligned}$$

3.1.3 Propagation of Relations

Road Network (7): The road network is composed, in our particular representation, of elements which are roads and intersections. Roads are also split into other elements, which consist of road sections. Connections have already been defined between roads and intersections on the one hand, and between road sections on the other hand. A road section, however, can also be connected to an intersection. Indeed, the first and last sections of a road are connected to intersections. So, connections between the different tree levels can be created.

In some cases, it can be interesting to *propagate* a relation established between two attribute-objects to their descendants or parents.

Let us clarify this for the connection relation. If two attribute-objects are connected, then their respective parents, if they are not the same attribute-object, are connected too. In the same way, if two connected attribute-objects have descendants, some of these descendants are also connected (between them and to their parent). Let us formalize this. But first, we need to elaborate on kinship relations.

The Notion of Kinship. $\Delta(\mathbf{A})$, which has already been introduced, is the set of attribute-subobjects that compose the attribute-object $\mathbf{A}$. In other words, $\Delta(\mathbf{A})$ is the set of descendants of $\mathbf{A}$[7].

Generalizing, $\Delta^i(\mathbf{A})$ is the set of descendants of $\mathbf{A}$ down to the i-th generation. For example, we have:

$$\Delta^1(\mathbf{A}) = \Delta(\mathbf{A}),$$
$$\Delta^{i+1}(\mathbf{A}) = \Delta^i(\mathbf{A}) \cup \left(\bigcup_{\mathbf{a} \in \Delta^i(\mathbf{A})} \Delta(\mathbf{a}) \right), \quad i \geq 2.$$

By extension, we have $\Delta^\infty(\mathbf{A})$ the set of all $\mathbf{A}$'s descendants in the considered representation.

Conversely, $\mathbf{P}^i(\mathbf{A})$ is the i-th parent of the object $\mathbf{A}$. It is itself an attribute-object of Ω. $\mathbf{P}(\mathbf{A})$ is the immediate parent of $\mathbf{A}$[8]. For example, we have:

$$\mathbf{P}^0(\mathbf{A}) = \mathbf{A},$$
$$\mathbf{P}^1(\mathbf{A}) = \mathbf{P}(\mathbf{A}),$$
$$\mathbf{P}^2(\mathbf{A}) = \mathbf{P}(\mathbf{P}(\mathbf{A})).$$

We also define $\mathbf{P}^i(\mathbf{A}_1, \mathbf{A}_2)$ as the i-th common parent to $\mathbf{A}_1$ and $\mathbf{A}_2$. In particular, we have $\mathbf{P}^1(\mathbf{A}_1, \mathbf{A}_2)$, the first common parent to $\mathbf{A}_1$ and $\mathbf{A}_2$.

The Propagated Connections. Let $\mathbf{a}_1$ and $\mathbf{a}_2$ be two attribute-objects. If $\mathbf{a}_1$ and $\mathbf{a}_2$ are connected, then, if they have descendants, some of these descendants are also connected (between them and to their parent, $\mathbf{a}_1$ or $\mathbf{a}_2$):

$$\left.\begin{array}{r} (\mathbf{a}_1, \mathbf{a}_2) \in \Gamma_{nc} \\ \Delta^\infty(\mathbf{a}_1) \neq \emptyset \\ \Delta^\infty(\mathbf{a}_2) \neq \emptyset \end{array}\right\} \Rightarrow \exists (\mathbf{a}'_1, \mathbf{a}'_2) \in (\Delta^\infty(\mathbf{a}_1) \cup \{\mathbf{a}_1\}) \times (\Delta^\infty(\mathbf{a}_2) \cup \{\mathbf{a}_2\}) \mid (\mathbf{a}'_1, \mathbf{a}'_2) \in \Gamma_{pc},$$

where Γ_{nc} is the graph of "natural" connections (*i.e.* already existing) and Γ_{pc} is the graph representing the propagated connections.

Let us write the reciprocal. Let us consider two attribute-objects $\mathbf{A}_1$ and $\mathbf{A}_2$ that have common parents, and let their first common parent, $\mathbf{P}^1(\mathbf{A}_1, \mathbf{A}_2)$, be $\mathbf{A}_1$'s i-th parent and $\mathbf{A}_2$'s j-th parent, *i.e.*

$$\mathbf{P}^1(\mathbf{A}_1, \mathbf{A}_2) = \mathbf{P}^i(\mathbf{A}_1) = \mathbf{P}^j(\mathbf{A}_2).$$

[7] When $\mathbf{A}$ belongs to a particular attribute A_k, the set is noted $\Delta_k(\mathbf{A})$.

[8] When $\mathbf{A}$ belongs to a particular attribute A_k, the parent is noted $\mathbf{P}_k(\mathbf{A})$.

Then, a necessary condition for $\mathbf{A}_1$ and $\mathbf{A}_2$ to be connected (by the propagated connections) is:

$$(\mathbf{P}^{i-1}(\mathbf{A}_1), \mathbf{P}^{j-1}(\mathbf{A}_2)) \in \Gamma_{cn}(\mathbf{P}^1(\mathbf{A}_1, \mathbf{A}_2)). \tag{3}$$

If condition (3) is not satisfied (which is the case for attribute-objects $\mathbf{A}_1$ and $\mathbf{A}_3$ of Figure 4), then the attribute-objects cannot have propagated connections between them.

If $\mathbf{A}_1$ and $\mathbf{A}_2$ are linked by a propagated connection, then all the pairs in which they are involved as well as their respective parents until their first common one are also linked by a propagated connection:

$$(\mathbf{A}_1, \mathbf{A}_2) \in \Gamma_{cg} \Rightarrow \forall (k,l) \in [0\,;i-1] \times [0\,;j-1] \setminus \{(0,0),(i-1,j-1)\} \quad (\mathbf{P}^k(\mathbf{A}_1), \mathbf{P}^l(\mathbf{A}_2)) \in \Gamma_{cg}.$$

Figure 4 shows an example of down propagation.

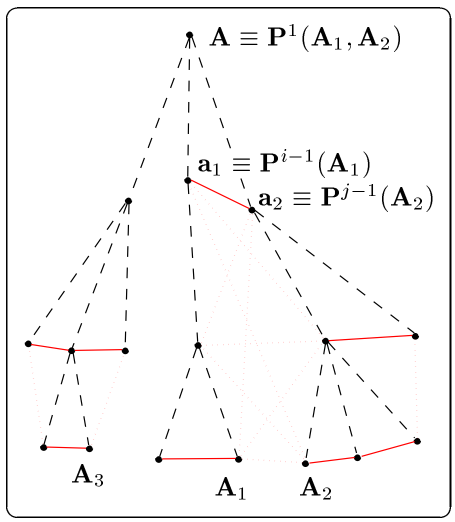


Figure 4: The propagated connection (top-down). The continuous lines show the established connections while the dotted lines show the propagated connections. The connections have been propagated down to attribute-objects $\mathbf{A}_1$ and $\mathbf{A}_2$, since a parent of one of the two is connected to a parent of the other. In contrary, no connection has been propagated between attribute-objects $\mathbf{A}_1$ and $\mathbf{A}_3$ because no parent of the first is connected to any parent of the second.

Representation (7.a): A road section is connected to an intersection if its parent (a road) is connected to the intersection. On the other hand, if a road is connected to an intersection, then there is a road section which is connected to the intersection.

The relations between attribute-objects can thus be propagated toward their descendants or parents. This can be useful in numerous cases, for example when:

- the existence of relation has not already been established between two attribute-objects;
- the information must be represented in a compact way (*e.g.* because of a limited memory) and extracted only when needed.

3.1.4 The Global Structure

The global structure, ${}^{\mathcal{R}}\mathcal{S}(\mathbf{O})$, of physical object $\mathbf{O}$ in representation $\mathcal{R}$ is composed of the set of attribute-objects decomposition trees and of the set of graphs representing the relations between attribute-objects of all attributes:

$$
{}^{\mathcal{R}}\mathcal{S}(\mathbf{O}) = \left(\bigcup_{k\in[0\,;N]} \Theta(\Omega_k), \bigcup_{k\in[0\,;N]} \Gamma(\Omega_k) \right).
$$

3.2 The Quantification

Road Network (8): We have to specify the road network dimensions, as well as traffic values, speed limits, pavement state, *etc.* For the geometry, for example, specifying dimensions is done by attributing to roads their length, to the intersections their position, *etc.*

The quantification of object $\mathbf{O}$ being represented is composed of a set of applications which quantify the attribute-objects of the different attributes of the object and a set of applications that quantify relations between these attribute-objects. The quantification of an attribute-object of a particular attribute is presented in Section 3.2.1, while the relations between attribute-objects quantification is presented in Section 3.2.3.

3.2.1 The Quantification of an Attribute-object A of an Attribute A_k

Road Network (9): In our representation, each road has a length and each intersection has an absolute position. Each road section is represented as a ribbon, *i.e.* a parametric curve representing the section's geometry in the terrestrial system of reference and the ribbon's width in each curve point.

The quantification of an attribute-object $\mathbf{A}$ is an application, which we note $\mathrm{Q}_k(\mathbf{A})$: $\mathrm{Q}_k(\mathbf{A}) \in \mathcal{F}$. It can be noted Q_k when there is no possible confusion.

Representation (9.a): A road is quantified by a constant application, which simplifies to the real number L, the road's length.

In the same way, an intersection is quantified by a constant application that gives a triplet of $\mathbb{R}^3$, (x, y, z), the intersection's position.

A road section is quantified by a parametric curve (whose parameter is s, the curvilinear abscissa, in the real interval $[0\,;L]$, L being the section road length) which provides a position for each section point and a width for each ribbon point:

$$
\mathrm{Q}_0(\mathrm{t}) = (\Gamma, [0\,;L], \mathbb{R}^4).
$$

The graph associates to each value s a quadruplet (x, y, z, l):

$$
\begin{aligned}
\mathrm{Q}_0(\mathrm{t})\colon [0\,;L] &\to \mathbb{R}^4 \\
s \in [0\,;L] &\mapsto (x, y, z, l).
\end{aligned}
$$

It is possible that the attribute-object $\mathbf{A}$ itself has descendants which can be quantified. Section 3.2.2 will show that $\mathrm{Q}_k(\mathbf{A})$ is not independent of $\mathrm{Q}_k(\mathbf{a})$, $\mathbf{a} \in \Delta_k(\mathbf{A})$.

3.2.2 Propagation of Attribute-objects Quantifications

Road Network (10): The road length information is contained in the road sections' parametric curves: it is the sum of all length of projected curves on the terrestrial system of reference (the curve without the width information). In the same way, the total road network length can be calculated from the roads' length while it's area can be calculated from the intersections' convex hull (at least in a first approximation).

In a given attribute's hierarchy, a similar quantitative information must be present in two consecutive levels. This means that the information represented by the quantification of an attribute-object at one level must be contained in the information represented by the it's descendants' quantification. This does not mean that the information must be present under the same form at both levels, but that the information present at the raw level must be extracted from the information at the finer level. In other words, the quantification of an attribute-object $\mathbf{A}$ must be expressible as a function of the quantifications of the attribute-objects $\mathbf{a}$ from $\Delta(\mathbf{A})$.

An application Q_p representing this propagated quantification for the attribute-object $\mathbf{A}$ can thus be extracted. Its domain is the set of attribute-subobjects' quantifications values. Its values represent the possible "values" of the attribute-object:

$$\mathrm{Q}_{p,k}(\mathbf{A}) \in \mathcal{F}\left(\bigcup_{\mathbf{a}\in\mathbf{A}} \mathrm{pr}_2\mathrm{Q}_k(\mathbf{a}), A\right),$$

where $\mathrm{Q}_{p,k}$ is a propagated quantification for the A_k attribute.

Representation (10.a): The complete road network will be quantified by its total length L. It is the sum of all its composing roads length:

$$\mathrm{Q}_0(\mathrm{R}) = (\Gamma, \mathtt{L}, \mathbb{R}),$$

where $\mathtt{L}$ is an entity representing the total road network length (see Representation 9.a). The graph is described by:

$$L = \sum_{\mathrm{r}\in\Delta(\mathrm{R})} l(\mathrm{r}),$$

where $l(\mathrm{r})$ is the road length:

$$l(\mathrm{r}) \in \mathrm{pr}_2\mathrm{Q}_0(\mathrm{r}).$$

3.2.3 The Relations' Quantification

Road Network (11): All the relations introduced until now where qualitative. But some quantitative relations may exist, *e.g.* the shortest distance, following the roads, between intersections. This relation, in addition to being represented by the relations graph (Road Network 6), must be quantified.

The relations between attribute-objects of Ω_k can be quantified (but not necessarily, if the relation is qualitative, for example) by associating an application to them. But this application must not be the same for all relation instances. For example, the relation "is at x meters from" where x is the relation quantifier can be applied to all attribute-objects of

Ω_k with a different quantification for each arc. This implies that each arc is labeled with a set of triplets constituted by one element of R, the set of relations labels, one element of $\mathbb{N}$, to distinguish eventual multiples attribute-objects represented by *one* node (Equation (2)), and one element of the applications set. The labeling function F_k for the graph representing the relations between attribute-objects of Ω_k is thus written:

$$\mathrm{F}_k\colon \Gamma(\Omega_k) \to \mathcal{P}(R \times \mathbb{N} \times \mathcal{F} \cup \{\emptyset\}). \tag{4}$$

In the previous equation, $\mathcal{F} \cup \{\emptyset\}$ is here to handle relations that are not quantified.

Representation (11.a): Thus, we have to add to the relations graph (Representation 6.a) the arcs representing distance between intersections. On the other hand, the following application is associated to the relations graph:

$$\mathrm{F}_0\colon \Gamma(\Omega_0) \to \mathcal{P}(\{c, n, s, e, o, d\} \times \mathcal{F} \cup \{\emptyset\}),$$

where $\{c, n, s, e, o, d\}$ is the set of labels representing the various geometric attribute relations. In the previous equation, the set $\mathbb{N}$ in Equation (4) is not necessary here because each geometric decomposition tree's node represent only one geometrical attribute-object. The application F_0 establishes a correspondence between each arc representing the connection and $(c, \emptyset)$, the relation being qualitative, and between cardinal relations ("is south to," "is north to," *etc.*) and similar pairs. To the arcs representing the distance relation, the application F_0 gives (d, D) where D is the constant application giving the distance between to intersections.

3.2.4 The Quantification of the Object O

The global quantification ${}^{\mathcal{R}}\mathcal{Q}(\mathbf{O})$ of the physical object $\mathbf{O}$ in the representation $\mathcal{R}$ is made of the set of the quantifications of all the attribute-objects for all $\mathbf{O}$'s attributes and of the set of quantifications of relations between attribute-objects for all attributes:

$$ {}^{\mathcal{R}}\mathcal{Q}(\mathbf{O}) = \left(\bigcup_{\substack{k\in[0\,;N]\\ \mathbf{A}\in\Omega_k}} \mathrm{Q}_k(\mathbf{A}), \bigcup_{k\in[0\,;N]} \mathrm{F}_k \right).$$

3.3 Inter-attribute Relations

Road Network (12): When people drive vehicles on a road network, they have to anticipate[(9)]. Thus they have to know what comes next to the place where they are on the network. For this purpose, the relations "is north to," "is south to," "is east to," and "is west to" can be expressed between road sections and attribute-objects of different attributes. Hence, if the vehicle's position and direction are known, then the traffic or pavement state of following road sections can also be known.

Relations between attribute-objects of a given attribute have already been introduced. We also have to define inter-attributes relations. Indeed, although attributes have been

[(9)]No, it is not a driving instruction manual!

represented independently, they are not independent. An interesting relation expresses correspondences between attribute-objects, attribute-objects' quantification applications and values of these applications for different attributes. This relation says, for example, that an object part is red while another one is blue or gives the density equation of the different object's parts.

Road Network (13): For example, we must be able to get the traffic value on a road at a raw level and on each section road point at a fine level (see Road Network (3) and (4)). This is also true for the pavement state. When this state is available, we must know the "travellability," an attribute which gives the maximum speed a car can travel depending on the pavement state and the road geometry but not depending on the speed limitations. Thus, we have to establish correspondences between road points and traffic values, road points and pavement states, and travellability values, pavement states, and road points.

The inter-attributes relations between two attributes are thus represented by a bipartite graph. The two sets from which pairs of nodes are made contain the attribute-objects, the quantifications of these attribute-objects, and the values these quantifications take for the two attributes. The set of quantitative values taken by the attribute A_k is noted V_k:

$$\mathrm{V}_k = \bigcup_{\mathbf{A}\in\Omega_k} \mathrm{pr}_2\mathrm{Q}_k(\mathbf{A}).$$

In the same way, the set of quantifications of all the attribute-objects in Ω_k is noted Ξ_k:

$$\Xi_k = \{\mathrm{Q}_k(\mathbf{A}) \mid \mathbf{A} \in \Omega_k\}.$$

The graph Γ_{ij} representing the inter-attributes relations between the two attributes A_i and A_j is thus written:

$$\Gamma_{ij} = \Gamma(\Omega_i \cup \Xi_i \cup \mathrm{V}_i\,,\, \Omega_j \cup \Xi_j \cup \mathrm{V}_j),$$

with $i \neq j$. As for all the relations presented so far, the inter-attributes relations can be quantified (the relations expressing correspondences are qualitative). A labeled graph is thus built with the previous graph Γ_{ij} by associating to it the following function F_{ij}:

$$\mathrm{F}_{ij}\colon \Gamma_{ij} \to \mathcal{P}(R_{ij} \times \mathbb{N} \times \mathcal{F} \cup \{\emptyset\}), \tag{5}$$

where R_{ij} is the set of labels of relations between attributes i and j and $\mathbb{N}$ is here for the same reason as in Equation (2).

The set of all graphs representing all inter-attributes relations for all object $\mathbf{O}$'s attributes pairs in the representation $\mathcal{R}$ is noted ${}^{\mathcal{R}}\mathcal{I}(\mathbf{O})$:

$${}^{\mathcal{R}}\mathcal{I}(\mathbf{O}) = \bigcup_{\substack{i,j=0,N\\ i\neq j}} \Gamma_{ij}.$$

Let us note that ${}^{\mathcal{R}}\mathcal{I}(\mathbf{O})$ is a multi-partite graph. To quantify the inter-attributes relations, the graph is labeled with the function F which is written:

$$\mathrm{F}\colon {}^{\mathcal{R}}\mathcal{I}(\mathbf{O}) \to \mathcal{P}(R \times \mathbb{N} \times \mathcal{F} \cup \{\emptyset\}), \tag{6}$$

where R is the set of all inter-attributes relations for all $\mathbf{O}$'s attributes.

The function described in (6) is in fact the "reunion" of functions (5) for all attributes pairs, *i.e.* the domain of the first is the reunion of the domains of the seconds while they all take their values in the same set.

An other remark must be made. We only speak of inter-attributes relations between *two* attributes. However, relations between more than two attributes are possible and their definition is similar.

4 A Simple Example

In order to illustrate GNOME, a simple (and essentially graphical) example is proposed. The goal is to represent a car for, *e.g.*, a recognition task. Two attributes are represented: geometry and color at three levels of detail which are shown in Figure 5.

4.1 The Structure

Let us first consider the geometric structure. The car is represented by a rectangular parallelepiped (pr_1) at a first level, is split in two rectangular parallelepiped (pr_2 for the body and pr_3 for the cockpit) and four cylinders (one for each wheel). The wheels are identical, except for their position. Thus, they will be represented by only one attribute-object (c_1) in $\Delta_k(\mathbf{A})$. The decomposition arc which links pr_1 to c_1 is quantified by the value 4 (Figure 6). The geometric attribute-object "wheel" is itself split into a rim, represented by a cylinder (c_2), and a tire, represented by a torus (t). The cockpit is split into a rectangular parallelepiped (pr_4) and two triangular prisms (pt_1 and pt_2). Figure 6 shows the different geometrical attribute-objects and their decomposition and connection relations.

The color's attribute-objects are points in the RGB space. There is a point for each color on the car. The wheels' color at the raw level is an average color of the tire's and rim's color. This is the same thing for the rectangular parallelepiped representing the car at the rawer level. The corresponding structure is shown in Figure 7(a). There is no other relation than the decomposition.

4.2 The Quantification

The quantification of the geometrical attribute-objects is made by associating an application to the rectangular parallelepiped, cylinders, triangular prisms, and tori. This application specifies the dimensions of each part. For example, for a rectangular parallelepiped, the application can be written:

$$\begin{aligned} \mathrm{Q}(pr)\colon \{(\mathtt{L},\mathtt{l},\mathtt{h})\} &\to \mathbb{R}^3 \\ (\mathtt{L},\mathtt{l},\mathtt{h}) &\mapsto (L,l,h), \end{aligned} \tag{7}$$

where $\mathtt{L}$, $\mathtt{l}$, and $\mathtt{h}$ are entities representing the rectangular parallelepiped's dimensions and L, l, and h are the dimensions. This a way to formalize a constant application.

Quantifying the color's attribute-objects is made by locating them in the RGB space (Figure 7(b)).

4.3 The Inter-attribute Relations

A single inter-attributes relation is represented: the correspondence between geometry and color. It gives the color of the different geometrical parts by associating to each geometrical attribute-object a color attribute-object. The graph representing this relation is shown in Figure 8.

5 Comparison With Other Models

We propose now a comparison between different models presented in the literature and GNOME. These models are the ones from Marr and Nishihara ([Mar82, Nis81]), Brooks ([Bro81]), Herman and Kanade ([HK84]), and Allen ([All88]).

All these models represent only the objects' geometry. In addition to the geometry, GNOME also represents other (non-geometric) attributes.

5.1 Marr and Nishihara's Model ([Mar82, Nis81])

This model uses generalized cones as representation primitives. A coordinate system is associated to each cone. The *3-D model* of an object is composed of two parts: the first part is the *model axis*, *i.e.* a cone giving the object's raw form and orientation, while the second part consists of the object's *component axes*, *i.e.* a set of cones, each giving the raw form and orientation of the different object's components. All the object's components mush have similar sizes. Each component is itself an object and, as such, can have a 3-D model. The components' axes are positioned with respect to the model axis or one of the components axes (in this case, this axis is the *principal axis* and is chosen as touching or as being the nearest to a maximum number of components axes; it is positioned with respect to the model axis). The components' size is also given relatively to the model axis. The positions and sizes can be given precisely or in an inexact manner in order to have a variable object structure.

Thus, there is a structure of decomposition tree which is very similar to GNOME's one. However, the tree's nodes *are* generalized cones (with a system of reference associated to them), thus integrating the notions of structure and quantification of GNOME (see Sections 3.1 and 3.2). On the other hand, there is only one possible representation primitive(10). A relation between objects (or, more precisely, between object's components) is represented in the Marr and Nishihara's model: the relative spatial position. This relation is quantified in a manner similar to the GNOME's quantification (however, without speaking of application). However, the relation is applied only to components of a given object and is thus not propagated to other tree levels (see Section 3.1.3). This clearly shows a difference between the two models. In Marr and Nishihara's one, the object and its components are at the same level in the tree, while in GNOME, they are at different levels. There are several ways in GNOME to represent the positioning between an attribute-object and its descendants. For example, the relation can be propagated (see Section 3.1.3) between an attribute-object and its descendants. It is also possible to quantify the attribute-object's descendants by associating

(10)Marr ([Mar82]) says that an extension to surface primitives is possible.

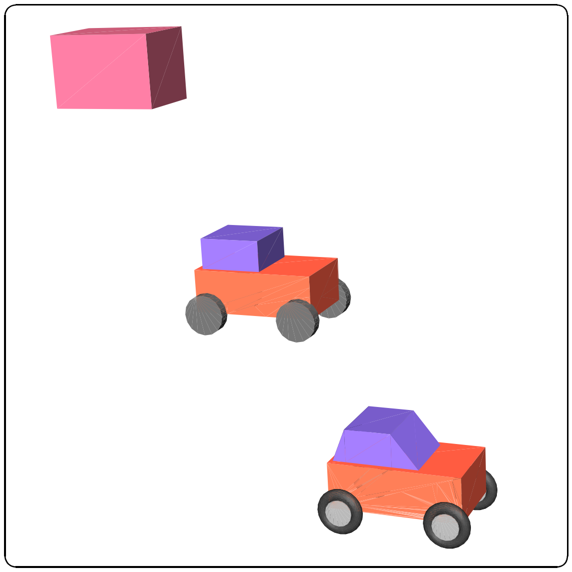


Figure 5: A car at three levels of representation

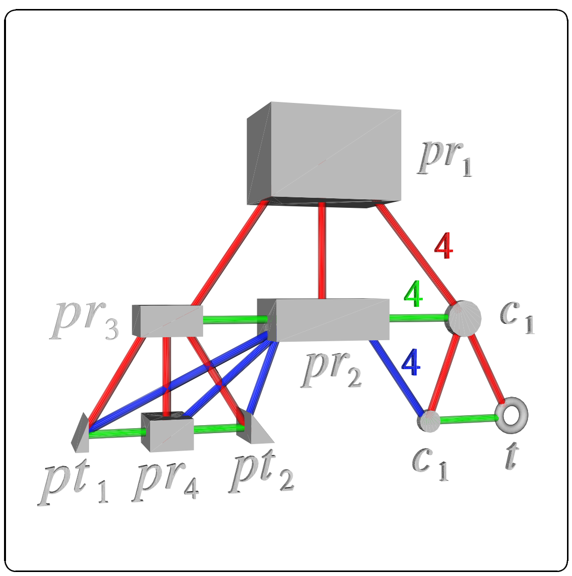
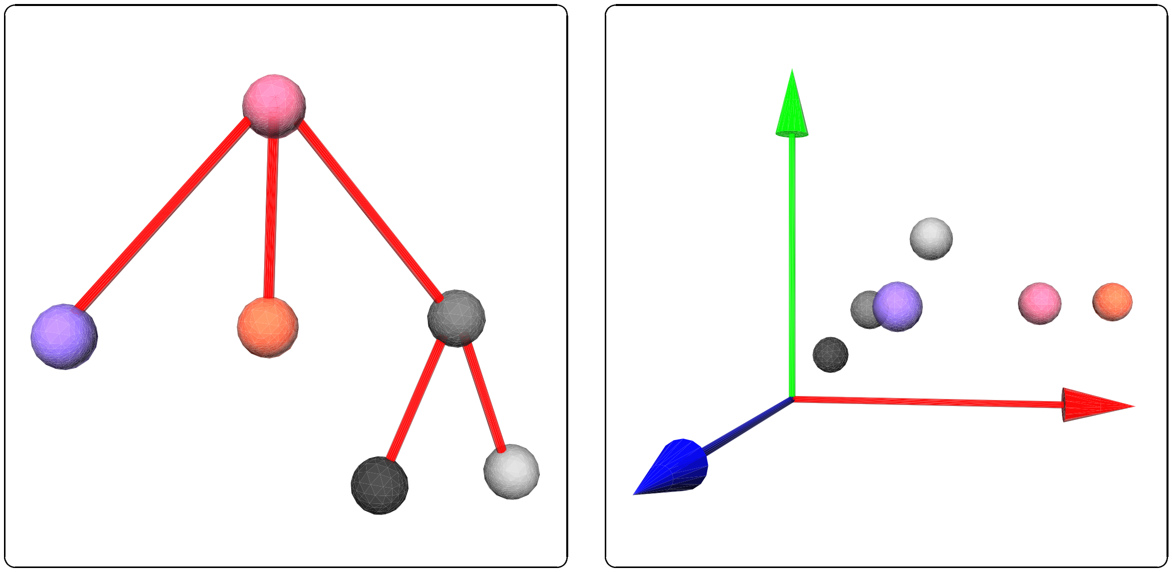


Figure 6: The cars decomposition tree and its connections. The decomposition is shown by red arcs while connections are represented by green and blue arcs. The blue arcs link attribute-objects from different levels and, as such, may be propagated connections from green arcs. The digits "4" give the number of attribute-objects in relation, as described in Equations (1) and (4) (for clarity reasons, the digits "1" are not shown on the drawing).



(a) The color's structure

(b) The quantification of color's attribute-objects

Figure 7: The color's structure and the quantification of its attribute-objects

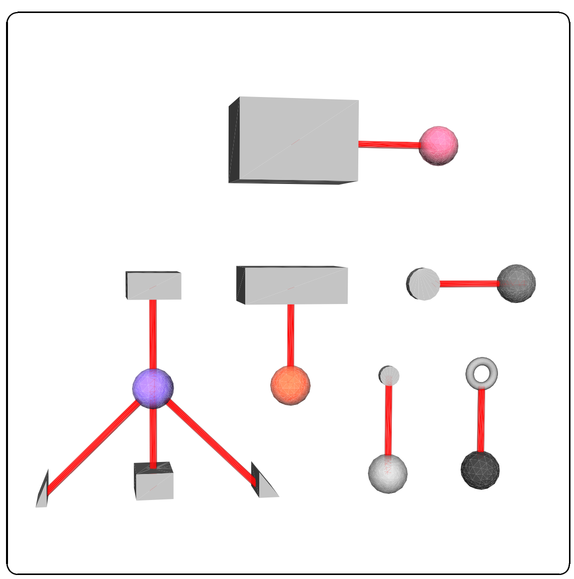


Figure 8: The graph representing the correspondence between geometry and color

to them a primitive (here a generalized cone) whose equation is given in the attribute-object system of reference.

In Marr and Nishihara's model, each representation part (*description* in their terminology) is a 3-D model completely independent. This means that a process (of recognition in this case) deals at a time with only a little part of the representation it uses. In GNOME, this can be the case too, each little part being a subset of Ω_0 (or more generally Ω_k) corresponding to a level of one branch of Θ_0 (or Θ_k) with the subgraph of the corresponding relations(11).

5.2 Brooks' Model ([Bro81])

In his system ACRONYM, Brooks uses an object model made of two graphs: the *objects graph* and the *restriction graph*. This model deals only with geometry.

The objects graph contains the objects' volumetric description as well as the spatial relations between the different objects' parts. The graph's nodes are objects or objects' parts. The objects are geometrically represented by generalized cylinders. In GNOME, there is no predetermined geometric primitive; different primitives can cohabit. They are defined by attribute-objects' quantifications (Section 3.2.1) *e.g.* with parametric equations. The graph's (directional) arcs express the *subpart* and *affixment* relations. The subgraph (of the objects graph) defined by the subpart relation is a tree arranged in a hierarchy of objects' level of details. The subpart relation gives connection between objects and, thus, this tree does not correspond to the (hierarchical) *decomposition* tree of (attribute-)objects at different levels (for example, of detail) of GNOME but rather to the subgraph (of relations between attribute-objects graph) constructed from the descendants of an attribute-object and representing only connection. The hierarchy is thus not made in the same way in ACRONYM's model and GNOME. In the first, it is only matter of level of detail and in the second it is matter of decomposition (or "globalisation," depending on the view point). On the other hand, circular connections can exist in GNOME.

A reference system is associated to each object. The affixment arcs link these systems. The affixment is a possible relation of GNOME: "Apply translation t and rotation r to switch from this attribute-object to this other" where the values of t and r belongs to the relation's quantification.

Each entity manipulated in ACRONYM's model has slots whose filler quantifies the entity. In this way, an object has slots to quantify its representing cylinder. The subpart and affixment relations have a slot to give the number of related objects. For example, an airplane has two wings. Thus, there is an arc of subpart between object *fuselage* and object *wing* with the quantity 2 (the two wings being identical). When an affixment is applied to several objects (quantity greater than 1), then an index, whose value is limited by the quantity, is used to express the different affixments. The same process is done in GNOME (see Section 3.1.1 and Equation (1) as well as, for an example, Equation (7)).

If the objects' fillers can vary, then an object class is defined rather than a particular object. In the same manner, subpart and affixment classes can be defined. The slots will thus be filled with fillers. Next, constraints over fillers are imposed in order to define a class.

(11)This can also include a part of some (or all) other attributes if the process uses them.

If other constraints are added, a subclass is defined. This allows size variation with cylinders fillers, structure change with quantities associated to subpart and affixment arcs, and spatial relations variation with affixment fillers.

The restriction graph describes objects classes. Each node is an object class, *i.e.* a set of constraints to fillers for instances of this class. The (oriented) arcs of this graph show the specialization of a subclass relatively to a class, *i.e.* a constraints addition.

GNOME has no explicit way to represent classes and thus instances. However, variations in size, spatial relations (and even any relations), and structure are possible. Indeed, the attribute-objects' quantification can give an exact value, which corresponds to a class instance of ACRONYM, or only a constraint, under the form of an interval, a probability distribution, *etc.* This gives the variation in size. The relations can be quantified in the same manner, in particular the transformation relations. This gives the variation in spatial relations. For the structure variation, this is done by the quantification of the decomposition relation with a value which indicates the instance number of an attribute object (see Section 3.1.1). This value can be "exact", for a rigid structure, or constrained (by an interval, a probability distribution, *etc.*), for a variable structure. Of course, if the structure is variable, the relations implicating the attribute-objects whose quantity is variable have to be quantified with an index indicating to which attribute-object the relation is applied (Equation (2)).

5.3 Herman and Kanade's Model ([HK84])

MOSAIC is a urban scenes reconstruction system. The objects modeled in this system are buildings, *i.e.* polyhedrons (objects made of faces, edges, and vertices). The model developed for MOSAIC is made of a graph of symbolic primitives, such as faces, edges, and vertices, with their topology and geometry. More precisely, the graph's nodes are of two types. On one hand, topological primitives, *i.e.* faces, edges, vertices, and groups of faces belonging to the same face, give the objects' structure (the scene). On the other hand, geometric primitives, *i.e.* plans, straight lines, and points, quantify the topological primitives and, thus, the objects. The graph's arcs are also of two types. The arcs "made of" represent the decomposition of topological primitives into other ones(12). The geometric constraints arcs link geometrical nodes to topological ones to quantify them. The nodes representing faces, edges, and vertices are labeled as *confirmed* or *not confirmed*, whether the element represented by the node is derived directly from the image or only inferred from it.

Thus, as in GNOME, there is a hierarchy created by the objects' decomposition. However, it is here a rigid decomposition due to the represented objects' nature. On the other hand, the decomposition does not correspond to a progressive increase in representation levels (of details), but rather to a reduction of information, contrary to GNOME. For example, the decomposition of an edge into two vertices only suppresses information. Indeed, the edge is constrained by geometric primitives consisting of two *point* and a *line*, while the two vertices are only constrained by two *points* (the same as for the edge).

On the other hand, this type of decomposition leads to a graph, while the GNOME's decomposition is representable only by a tree. Indeed, in MOSAIC's model, an edge can

(12)A face is made of a group of edges and edges are made of vertices.

belong to two different faces and a vertex can belong to several edges. The edges and vertices can also be seen as two types of faces connections. In this case, the MOSAIC's model graph is similar to a representation following GNOME in which the objects are split into faces connected by their contact edges or vertices (by means of a connection relation quantified by the edges or vertices by which the contact is made). An other difference is that the notion of geometric primitive does not exist in GNOME. These primitives are replaced by quantifications (see Section 3.2).

5.4 Allen's Model ([All88])

Allen proposes an object recognition system in which the objects' knowledge is stored in a model base. Each model represents, in a hierarchical manner and with a tree, a particular object. At the tree's root is the object itself. In the node representing it, there is its quantification as well as a pointer list toward the object's components and features (holes and cavities). The object's quantification is made of a bounding box and a complexity index. The following tree level is constituted of the objects components and features. Each node representing a component contains the component's quantification and a pointer list toward the component's surfaces. The quantification is a bounding box, an area, and a priority index. Each node representing a feature (hole or cavity) contains a quantification, which gives an area, priority, form, and system of reference. These nodes are tree's leaves. The third tree's level is constituted of surfaces forming the object's components. The nodes representing them contain the surface quantification and a pointer list toward bicubic patches forming the surface. The quantification is a bounding box, an area, informations about the surface closeness symbolic form (planar, cylindrical, or curve), and a system of reference (if one exists). The last level is the one of bicubic patches whose nodes contains their parametric description as well as values such as area, bounding box, normal vector, or symbolic form. Attached to the root, a constraint list describe the adjacency relations (connections) between the elements of each tree level.

There is thus a tree very similar to the decomposition tree in GNOME (Section 3.1.1): an object is split into components and features, the components are split into surfaces themselves split into bicubic patches. Each tree node is quantified (while this is not an obligation in GNOME). The descent corresponds to an augmentation of level of details and to an increase in information about the object, as it must be in GNOME. Moreover, the quantitative information at one level can be calculated from the one at the inferior level (see Section 3.2.2). For example, the area of a component is the sum of the areas of the surfaces composing it, the bounding box of a surface can be calculated from the patches' bounding boxes and positions. The connection relation (Section 3.1.2) is present while not propagated (as it can be in GNOME (Section 3.1.3)).

6 Conclusion

We have presented a generic object model in which objects are represented by their attributes. Each attribute leads to an arborescent structure of decomposition of attribute-objects (objects defined in the attribute representation space from groupings of its values) and a graph representing the relations existing between the attribute-objects. The attributes are represented in two parts: one is topological (or qualitative) while the other is quantitative. A graph makes links between the different attributes by representing inter-attributes relations.

Finally, a comparison between GNOME and several models presented in the literature has been presented. It shows that GNOME is comparable to these models with the advantage of not sharing some of their limitations concerning, for example, representation primitives or representable attributes and relations.

References

[All88] Peter K. Allen. Integrating vision and touch for object recognition tasks. *International Journal of Robotics Research*, 7(6):15–33, 1988.

[Big85] Norman L. Biggs. *Discrete Mathematics.* Clarendon Press, Oxford, 1985.

[Bro81] Rodney A. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17(1-3):285–348, 1981.

[HK84] Martin Herman and Takeo Kanade. The 3D MOSAIC scene understanding system: Incremental reconstruction of 3D scenes for complex images. In *Proceedings of DARPA Image Understanding Workshop*, pages 137–148, 1984.

[Lab95] Frédéric Labrosse. Modélisation de l'Objet et application à ses représentations. Technical Report GRPR-RT-9507, Groupe de Recherche en Perception et Robotique, École Polytechnique, Montréal, QC, October 1995.

[Mar82] David Marr. *Vision.* W. H. Freeman and Company, 1982.

[Nis81] H. Keith Nishihara. Intensity, visible-surface, and volumetric representations. *Artificial Intelligence*, 17:265–284, 1981.

List of symbols

$\Delta^{\infty}(\mathbf{A})$ The set of all $\mathbf{A}$'s descendants, page 8

$\Delta^{i}(\mathbf{A})$ The set of descendants of $\mathbf{A}$ down to the i-th generation, page 8

$\Delta_{k}(\mathbf{A})$ The set of the descendants of attribute-object $\mathbf{A}$ of the attribute A_k, page 6

$\Gamma(A)$ A graph whose arcs are elements of $A \times A$, page 2

$\Gamma(A, B)$ A (bipartite) graph whose arcs are elements of $A \times B$, page 2

$\Gamma(A_1, \cdots, A_n)$ A n-partite graph whose arcs are elements of $A_1 \times \cdots \times A_n$, page 3

$\Gamma(\Omega_k)$ The graph representing the relations between attribute-objects of A_k attribute, page 7

$\Gamma_r(A)$	A graph whose arcs are elements of $A \times A$ representing a particular relation r, page 2
Γ_{ij}	The inter-attributes relations graph between attributes A_i and A_j, page 13
Ω_k	The set of the attribute-objects which are the nodes of the decomposition tree fro the A_k attribute, page 6
$\Pi(\Gamma)$	A path in the graph Γ, page 3
$\Theta(A)$	A tree whose nodes are elements of A, page 3
$\Theta(\Omega_k)$	The decomposition tree of the A_k attribute's objects, page 6
A_k	The k-th attribute (A_0 being the geometric one), page 4
$\mathbf{A}$	A generic attribute-object, page 5
$\mathbf{a}$	A generic attribute-subobject, page 6
F	The labeling function of the inter-attributes relations graph for all attributes, page 13
F_k	The labeling function for the graph representing relations between attribute-objects of the attribute A_k, page 12
F_{ij}	The labeling function of the graph of (inter-attributes) relations between attributes A_i and A_j, page 13
${}^{\mathcal{R}}\mathcal{I}(\mathbf{O})$	The graph of inter-attributes relations of object $\mathbf{O}$ in the representation $\mathcal{R}$, page 13
$\mathbf{O}$	The physical object represented, page 5
$\mathbf{P}^i(\mathbf{A})$	The i-th parent of the object $\mathbf{A}$, page 8
$\mathbf{P}^i(\mathbf{A}_1, \mathbf{A}_2)$	The i-th common parent to $\mathbf{A}_1$ and $\mathbf{A}_2$, page 8
$\mathcal{P}(E)$	The set of parts of the set E, page 3
$\mathrm{pr}_1\Gamma$	The first projection of the graph Γ, page 3
$\mathrm{pr}_2\Gamma$	The second projection of the graph Γ, page 3
${}^{\mathcal{R}}\mathcal{Q}(\mathbf{O})$	The quantification of the object $\mathbf{O}$ in the representation $\mathcal{R}$, page 12
$\mathrm{Q}_k(\mathbf{A})$	The quantification of the attribute-object $\mathbf{A}$ of the attribute A_k, page 10
$\mathcal{R}$	A physical object representation, page 5
${}^{\mathcal{R}}\mathcal{S}(\mathbf{O})$	The global structure of attribute-object $\mathbf{O}$ in representation $\mathcal{R}$, page 10