GENETICA CAD:

Documentation of the Language G-CAD

© Lefteris Viririakis left@genetica-informatics.org

Contents

DOCUMENTATION OF THE LANGUAGE G-CAD

Introduction

G-CAD, which stands for GENETICA CAD, is a computer language specialized in problem solving within the framework of architectural designing. G-CAD is a GENETICA-like (non GP) language where data generation scenarios evolve with respect to either confirmation or optimization goals formulated in the G-CAD program. Design objects are represented in G-CAD as interrelated logical constructions having both internal properties and communication channels. G-CAD programming has the following features:

- Domain specific knowledge expressed in formal logic is encapsulated in design objects
- Objects exhibit context sensitive behavior based on object-to-object interactions.
- Both top-down and bottom-up interactions occur in the object hierarchy.
- Higher level objects result from behavior-driven selfassembly of lower level ones.

G-CAD has been developed in GENETICA under the principles presented in *GENETICA-based_Languages.PDF*. Specifically G-CAD has been implemented as a GENETICA program, having input an arbitrary G-CAD program. The execution of the latter program reflects the execution of the former one which operates as a G-CAD interpreter and it is controlled by the evolutionary computational system within GENETICA's environment.

Section 1 presents G-CAD's terms and objects, the latter viewed as composite terms. Section 2 presents both the syntax and the functionality of the G-GAD formulae. Section 3 presents the data organization in both the problem formulation and the solution. The Appendix A provides a reference on special classes of G-CAD formulae, while the Appendix B presents the formulation in G-GAD of a design problem concerning a hotel's floorplan.

Notation

- Bold words denote names or values of either variables or formulae. A quoted bold word denotes either a string value as it is or the name of a variable or a formula, whereas a non quoted bold word denotes either the value of the variable or the content of the formula named by the word, e.g. "**word**" denotes the string "word" which represents the name of a variable or a formula whereas **word** denotes the value of a variable named "word" or the content of a formula named "word".
- Numerical values followed by dot-zero denote real numbers. Values not followed by dot-zero denote integer numbers, e.g. **4.0** is a real number whereas **4** is an integer number.

1 G-CAD terms and objects

1.1 *General terms*

Atomic terms in G-CAD are integers, reals and symbols viewed as strings. An empty list is also an atomic term. If $\mathbf{t_1, \ldots t_n}$ are terms then the list $\mathbf{(t_1, \ldots t_n)}$ is also a term. This recursively defines as a term any tree-structured list whose terminals are atomic terms.

1.2 *Property Lists*

A "property list" is a list of pairs, where each pair has the form :
(property_name, property_value).

1.3 *Spatial Maps*

A Spatial Map represents a configuration of spatial attributes on the Cartesian Plane by mapping regions of the Cartesian Plane to property lists, where properties represent spatial attributes. The geometry of a Spatial Map is defined by two lists of real numbers which are called the "dimension lists". If **a** and **b** are successive reals in a dimension list then **a < b**. Each dimension list is associated to an axis of the Cartesian Plane while its elements represent coordinates on the axis. Each coordinate defines a line perpendicular to the axis. The lineframe defined by both dimension lists divides the Cartesian Plane into rectangular regions called "cells". The cells form rows and columns. The number of the cells in each row (respectively column) is **n+1** where **n** is the number of the lines perpendicular to the X (respectively Y) axis of the Cartesian Plane The first and the last cell of each row (respectively column) are considered to have infinite area since they lack one or two edges. Each cell is assigned a property list.

1.4 *Queries*

A Query "applied" on a property list, name it **PL**, is a list representing a logical statement on the content of **PL**. A Query has one of the following forms:

- **("log_eq", ($\mathbf{p_n}$, $\mathbf{p_v}$))**, where $\mathbf{p_n}$ is a property name and $\mathbf{p_v}$ is a value of any type, is confirmed if and only if $\mathbf{p_v}$ equals the value of a property named $\mathbf{p_n}$ in **PL**.
- **("log_ism", ($\mathbf{p_n}$, $\mathbf{p_v}$))**, where $\mathbf{p_n}$ is a property name and $\mathbf{p_v}$ is a value of any type, is confirmed if and only if the value of a property named $\mathbf{p_n}$ in **PL** is a list, name it **Lv**, while $\mathbf{p_v}$ is an element of **Lv**.
- **("log_and", ($\mathbf{L_1}$, ... $\mathbf{L_n}$))**, where $\mathbf{L_1}$, ... $\mathbf{L_n}$ are Queries applied on **PL**, is confirmed if and only if all $\mathbf{L_1}$, ... $\mathbf{L_n}$ are confirmed.
- **("log_or", ($\mathbf{L_1}$, ... $\mathbf{L_n}$))**, where $\mathbf{L_1}$, ... $\mathbf{L_n}$ are Queries applied on **PL**, is confirmed if and only if at least one of $\mathbf{L_1}$, ... $\mathbf{L_n}$ is confirmed.
- **("log_not", (L))**, where **L** is a Query applied on **PL**, is confirmed if and only if **L** is not confirmed.

Given a Spatial Map, a Query could specify a region on the Cartesian Plane: the region includes the cells whose property lists confirm the Query.

1.5 *Direction vectors*

The vectors specified by the angles $\mathbf{0^0}$, $\mathbf{90^0}$, $\mathbf{180^0}$ and $\mathbf{270^0}$, measured counterclockwise from the positive direction of the *X* axis of the Cartesian Plane, are represented by the reals **1.0**, **2.0**, **3.0** and **4.0**. These vectors, in the specific notation, will be referred to as the direction vectors.

1.6 *OCS (Object Coordinate System)*

A OCS is a design object's local coordinate system represented as a list having the form $\mathbf{(m, a, x_0, y_0)}$ where $\mathbf{(x_0, y_0)}$ is the origin point, **a** is a direction vector specifying the direction of the *X* axis of the OCS and **m** is a mirroring coefficient. **m** accepts the values **1.0** and **-1.0**, where **1.0** represents the not mirrored condition (the *Y* axis shows at the left side of the *X* axis) while **-1.0** represents the mirrored condition (the *Y* axis shows at the right side of the *X* axis). In the latter case the *X* axis is reversed i.e. it has the opposite direction of **a**. When the OCS of a design object is mirrored then the design object is mirrored too. All the parameters defining a OCS refer to the coordinate system of the Cartesian Plane. The latter, which will be referred to as the World coordinate system, is represented by the list **(1.0, 1.0, 0.0, 0.0)**.

1.7 *Primitives*

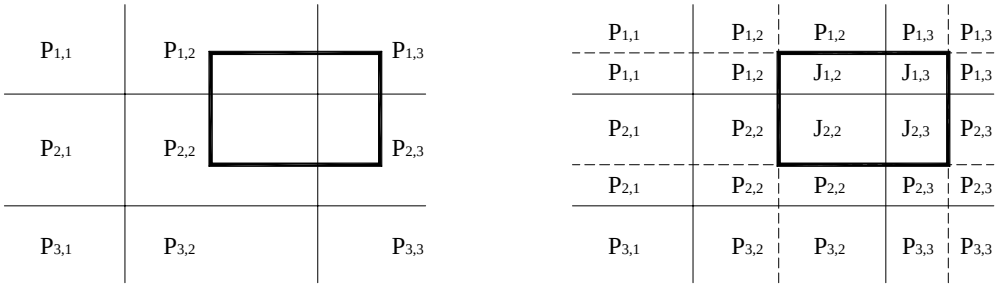
A Primitive is a design object, geometrically represented as a rectangle, which is formally represented as a list having the form **(1.0, PL, PCS, (X, Y))** where:

- **1.0** is the object type identity for a Primitive
- **PL** is a property list
- **PCS** is a OCS specifying the positioning of the Primitive on the Cartesian Plane: the lower left corner of the Primitive is positioned at the origin point of **PCS** while the lower and the left edge of the Primitive are positioned on the *X* and the *Y* axis of **PCS** respectively.
- **X** and **Y** (reals) are the dimensions of the Primitive towards the *X* and the *Y* axis of **PCS** respectively.

A Primitive can be "imprinted" to a Spatial Map, name it **SM**. The imprinting results to a transformation of **SM** defined by the following process:

- For each horizontal (respectively vertical) edge of the Primitive, which does not coincide with the boundary between successive rows (respectively columns) of **SM**, create a new boundary at that edge by dividing the row (respectively column) that includes the edge in two rows (respectively columns). Due to the division, each cell of the initial row (respectively column) is divided in two new cells, which share the same property list: this is the property list assigned to the initial cell.
- For each **PL**'s element, name it **Pair**, having property name **N** and property value **V**, and for each cell, name it **C**, included in the Primitive, apply the following revision process to the property list, name it $\mathbf{P_C}$, of **C**:
  - If $\mathbf{P_C}$ includes a property named **N** having a list value, name it $\mathbf{L_C}$, append **V** as an element to $\mathbf{L_C}$.
  - If $\mathbf{P_C}$ does not include a property named **N**, append **Pair** as an element to $\mathbf{P_C}$.

The procedure imprinting a Primitive to a Spatial Map is presented in Figure 1.



a) The Spatial Map before the Primitive's imprinting *b) The Spatial Map after the Primitive's imprinting*

Figure 1

The state of a Spatial Map, before a Primitive is imprinted, is presented at the left. Thin lines represent cell boundaries, while $\boldsymbol{P_{i,j}}$ *(i = 1, 2, 3, j = 1, 2, 3) are the property lists assigned to the cells. The Primitive is represented as a rectangle drawn with thick lines. The state of the Spatial Map after the Primitive's imprinting is presented at the right. Dashed lines represent new cell boundaries, while* $\boldsymbol{J_{k,h}}$ *(k = 1, 2, j = 2, 3) is the revision of the property list* $\boldsymbol{P_{k,h}}$ *caused by the Primitive's imprinting.*

1.8 *Units and Instances*

A Unit is a tree-structured design object whose terminals (leaf nodes) are Primitives. An Instance is a Unit's image which can be placed on the Cartesian Plane in different positions, orientations and mirroring conditions. Both Units and Instances can be represented as nodes in the tree-structure of a higher level Unit. The highest level Unit, i.e. the root Unit within the tree-structure, will be referred to as the World Unit. Every design object represented as a non-root node within the tree-structure of a Unit will be referred to as a sub-object of the Unit, while sub-objects represented as a Unit's child nodes will be referred to as child-objects of the Unit. Both Units and Instances can be "imprinted" to a Spatial Map. Imprinting a Unit or an Instance is equivalent to imprinting specific terminal Primitives of the Unit or the Instance depending on the call to the formula creating the Unit or the Instance respectively (see § 2.2.b, 2.2.c).

A Unit is represented as a list having the form $\boldsymbol{(2.0, PL_U, L_1, ... L_n)}$, where the elements $\boldsymbol{L_1, ... L_n}$ are optional, while an Instance is represented as a list having the form $\boldsymbol{(3.0, PL_I)}$. Within this representation the reals **2.0** and **3.0** are the object type identities for a Unit and an Instance respectively, $\boldsymbol{PL_U}$ and $\boldsymbol{PL_I}$ are the property lists of a Unit and an Instance respectively, and $\boldsymbol{L_i}$ (i = 1, … n) is a child-object of the Unit. Sub-objects include Primitives, Units (recursive definition) and Instances.

Within the Instance's representation, the information specifying both the Unit to be instanced and the geometric transformation that takes the instanced Unit to the Instance is registered in $\boldsymbol{PL_I}$ (see § 2.2.c).

1.9 *Positioning Lists*

Positioning Lists include specifications for the positioning of a design object on the Cartesian Plane, given the Spatial Map where the object is to be imprinted.

A Positioning List has the form $\boldsymbol{(QR_1, QR_2, L_a, L_m, v_{min})}$ where:

- **QR_1** and **QR_2** are Queries specifying two regions on the Cartesian Plane, name them **R_1** and **R_2** respectively. Consider the linear segments forming the boundary between **R_1** and **R_2** as vectors having **R_1** at the left side and **R_2** at the right side. These vectors will be referred to as the "positioning vectors". Name **L_v** the list of the positioning vectors.
- **L_a** is a list of direction vectors
- **L_m** is a list of mirroring coefficients (see § 1.6)
- **v_{min}** is a real

Let **L_{va}** be the list of the positioning vectors whose length is bigger or equal to **v_{min}** and whose direction is represented by an element of **L_a**. Each element, name it **v**, of **L_{va}** represents two potential positioning possibilities for the G-CAD object, depended on **L_m** (Figure 2):

- If **L_m** includes the value **1.0** then the G-CAD object can be positioned having the origin point of its OCS at the origin point of **v** and the *X* axis of its OCS having the direction of **v**.
- If **L_m** includes the value **-1.0** then the G-CAD object can be positioned having the origin point of its OCS at the end point of **v** and the *X* axis of its OCS having the opposite direction of **v**. In this case the G-CAD object should be mirrored with respect to the positioning vector, i.e. the *Y* axis of its OCS should be inversed.

*a) The mirroring coefficient has the value **1.0***

*b) The mirroring coefficient has the value **-1.0***

Figure 2

*Positioning of a G-CAD object with respect to a positioning vector at the boundary between the regions **R_1** and **R_2**. The value **1.0** of the mirroring coefficient results to the positioning presented at the left whereas the value **-1.0** results to the positioning presented at the right. In both cases, the small vectors named "**X**" and "**Y**" indicate the axes of the object's OCS.*

1.10 *Sizing lists*

A Sizing List includes specifications for the definition of the dimensions of a Primitive. It has the form **(F_X, V_X, F_Y, V_Y)** where **F_X** (respectively **F_Y**) is a one-element list including the name (viewed as a string) of a formula, referred to as the Dimension Definition Formula, which defines the length of the Primitive towards the *X* (respectively *Y*) axis of the Primitive's OCS, and **V_X** (respectively **V_Y**) is a list of values constituting the argument of the Dimension Definition Formula. The Dimension Definition Formulae are presented in Appendix A.1.

2 G-CAD formulae

2.1 *General features of G-CAD formulae*

G-CAD formulae have distinct input and output terms. Given the input terms the formula can be "called" i.e. the program represented by the formula can be executed. The output terms emerge as a result of the call.

The input of a G-CAD formula includes:

- a Spatial Map, name it $\mathbf{M_{inp}}$, referred to as the "input map".
- a pointer, name it $\mathbf{u_w}$, to the "World Unit" (see § 1.8).
- a pointer, name it $\mathbf{u_p}$, to a Unit, referred to as the "target Unit", which is either the World Unit or any Unit represented within the World Unit's structure.
- a vector value, name it $\mathbf{V_{inp}}$.

The output includes:

- a Spatial Map, name it $\mathbf{M_{out}}$, referred to as the "output map"
- a vector value, name it $\mathbf{V_{out}}$.

The input map represents the environment where a new design object is to be constructed, the World Unit represents the structure of all the design objects existing when the formula is called, the target Unit represents the existing design object where the new design object is to be appended as a child-object, and the output map represents the environment after the new design object's construction.

The terms $\mathbf{M_{inp}}$, $\mathbf{u_w}$, $\mathbf{u_p}$ and $\mathbf{M_{out}}$ are implicit in G-CAD, i.e. they do not appear in the G-CAD syntax. Both the implicit terms and their syntactical relations, which are opaque to the G-CAD programmer, are defined at run time given the data structure (described in § 3.1) that represents the problem statement. The input and the output section of a G-CAD formula, as the latter appears in the G-CAD syntax, include only symbols for the values included in $\mathbf{V_{inp}}$ and $\mathbf{V_{out}}$ respectively.

G-CAD includes formulae that create design objects (i.e. Primitives, Units or Instances). These formulae will be referred to as "primary formulae". Formulae that do not create design objects will be referred to as "secondary formulae". Design objects created as a result of a primary formula's call are appended to the target Unit as child-objects while they are imprinted at the input map of the call. The output map is the result of the imprinting. When a secondary formula is called the output map equals the input map of the call while the target Unit remains intact.

Atomic G-CAD formulae include both primary and secondary formulae. A high order atomic formula calls any other formula whose name appears in the high order formula's input. Non atomic G-CAD formulae include only formulae creating Units. The definition of a Unit creation formula will be referred to as a Unit Definition. Unit Definitions combine references to G-CAD formulae under G-CAD connectives. A G-CAD program is a list of Unit Definitions. References to G-CAD formulae are described in § 2.2 while Unit Definitions are described in § 2.3.

2.2 *References to G-CAD formulae*

Syntactically, a reference to a G-CAD formula is a list having the form **(t, F_N, S_{inp}, S_{out})** where **t** is a real number indicating the type of the formula, F_N is either the name of the formula (viewed as a string) or an empty list, depending on **t**, while S_{inp} and S_{out} are respectively the reference's input and the output section represented as lists of symbols, the latter viewed as strings. The process of calling the formula named F_N, given an input map, a target Unit and a complete value assignment for S_{inp}, which assigns the output values of the call to the respective symbols in S_{out}, will be referred to as the execution of the reference. References to G-CAD formulae are presented here.

a) Reference to the formula creating a Primitive

(1.0, ( ), (PL, Pos, SL), (X, Y, PCS)) where :

- **PL** is a symbol for the Primitive's property list (see § 1.7).
- **Pos** is a symbol for the Primitive's Positioning List (see § 1.9).
- **SL** is a symbol for the Primitive's Sizing List (see § 1.10).
- **X** is a symbol for the Primitive's dimension towards the *X* axis of the Primitive's OCS (see § 1.6).
- **Y** is a symbol for the Primitive's dimension towards the *Y* axis of the Primitive's OCS
- **PCS** is a symbol for the Primitive's OCS

The formula creating a Primitive is confirmed if and only if the Positioning List assigned to **Pos** indicates that the Primitive to be created can be positioned, given the input map (see § 2.1), while the Primitive is included in the input map's region specified by the Query constituting the first element of the Positioning List (see § 1.9).

If the Primitive is to be included a sub-object in an instanced Unit then the Primitive's property list should necessarily include a property named **"APA"** (which stands for "Acceptable Positioning Area") whose value is a Query. Given a Spatial Map where the instanced Unit is to be imprinted, the **APA** Query defines the region where the instanced Primitive should be included.

b) Reference to the formula creating a Unit

(2.0, F_N, S_{inp}, S_{out}) where :

- **F_N** is the name of the formula
- **S_{inp}** is a list of symbols for the input terms of the formula.
- **S_{out}** is a list of symbols for the output terms of the formula.

The confirmation condition for a formula creating a Unit depends on the formula's definition in the G-CAD code (see § 2.3), since a formula creating a Unit is a non atomic formula.

c) Reference to the formula creating an Instance

(3.0, ( ), (Pos, Q_{IUS}, P_{FLT}, IL), (IU_P, ICS)) where :

- **Pos** is a symbol for the Instance's Positioning List (see § 1.9).
- **Q_{IUS}** is a symbol for a Query which is applied to the property lists of all the Units that constitute World Unit's sub-objects, in a depth first order with respect to the World

Unit's tree-structure. The first Unit that confirms the Query is the instanced Unit.

- $\mathbf{P_{FLT}}$ is a symbol for a Query which is applied to the property lists of all the Instance's terminal Primitives. The output map of the formula call that creates the Instance emerge as the result of imprinting to the input map all the terminal Primitives whose property list confirms the Query.
- **IL** is a symbol for a property list referred to as the "Instance List". The input map's cells occupied by the Instance should have list-valued properties homonymous with the properties in the "Instance List". The value of each property in the "Instance List" will be appended as an element to the list-value of the homonymous property of each output map's cell occupied by the Instance.
- $\mathbf{IU_P}$ is a symbol for the pointer to the instanced Unit.
- **ICS** is a symbol for the Instance's OCS.

The formula creating an Instance is confirmed if and only if the Positioning List assigned to **Pos** indicates that the Instance to be created can be positioned, given the input map (see § 2.1), while each terminal Primitive of the Instance is included in the input map's region specified by the **APA** Query of the Primitive (see § 2.2.a). In the case of confirmation the Instance is defined to be the list $\mathbf{(3.0, PL_I)}$ (see § 1.8) where the property list $\mathbf{PL_I}$ includes the properties named "**IUS**", "**UCS**", "**IUP**", and "**l**", respectively having values the value assigned to $\mathbf{Q_{IUS}}$, the Instance's OCS, the pointer to the instanced Unit, and the "Instance List".

The transformation that takes the instanced Unit to the Instance is the transformation that takes the instanced Unit's OCS to the Instance's OCS.

d) Reference to a secondary formula

$\mathbf{(4.0, F_N, S_{inp}, S_{out})}$ where :

- $\mathbf{F_N}$ is the name of the referenced formula
- $\mathbf{S_{inp}}$ is a list of symbols for the input terms of the formula.
- $\mathbf{S_{out}}$ is a list of symbols for the output terms of the formula.

G-CAD secondary formulae are presented in the Appendix A.

e) Reference to the high order formula

$\mathbf{(5.0, (\ ), (F_{ref}), S_{out})}$ where :

- $\mathbf{F_{ref}}$ is a symbol for a list constituting a formula reference. This list will be referred to as the "input reference".
- $\mathbf{S_{out}}$ is a list of symbols for the output terms of the formula referenced in the "input reference".

The reference to the high order formula is equivalent to the "input reference", while the output values of the referenced formula are assigned to the homologous symbols in $\mathbf{S_{out}}$.

2.3 *Unit Definitions*

Formally a Unit Definition, i.e. the definition of a formula creating a Unit, is a list having the form:

(F_N, (Con, Inp, PL, Refs, Outp))

where:

F_N	is the formula's name, viewed as a string
Con	is a G-CAD connective
Inp	is a list of names of the formula's input terms, viewed as strings
PL	is a property list
Refs	is a list of G-CAD formula references (see § 2.2).
Outp	is a list of names of the formula's output terms, viewed as strings.

Inp, **PL**, **Refs** and **Outp** will be referred to as the input section, the property section, the reference section and the output section of the Unit Definition respectively.

Each symbol in the input section of a reference within **Refs** should be included in either **Inp** or in **PL**, as property names, or in the output section of a previous reference within **Refs**.

Each symbol in **Outp** should be included in the output section of a reference within **Refs**.

When the Unit creation formula is called:

1. A new Unit, name it **U**, is initialized as a list having the form **(2.0, PL_U)**, where PL_U is the Unit's property list (see § 1.8) which is initialized as the union of two property lists:

   a) the property list that includes the names within the input section of the Unit Definition as property names and the values assigned to these names as property values
   b) the property section of the Unit Definition (i.e. the property list **PL**).

   Either the property list (a) or the property list (b) should include a property named "**ID_UCS**" and a property named "**FLT**":

   - **ID_UCS** is a Query which is successively applied to the property lists of the Unit's sub-objects in a depth first order with respect to the Unit's tree-structure. The OCS of the first sub-object that confirms the Query becomes the OCS of the Unit and constitutes the value of another property named "**UCS**" which is automatically defined and appended to PL_U.
   - **FLT** is a Query which is applied to the property lists of all the Unit's terminal Primitives. The output map of the Unit creation formula is affected only by the terminal Primitives that satisfy the **FLT** Query.

2. **U** is appended as an element at the end of the target Unit of the Unit creation formula call
3. Specific references within **Refs** are executed. Both the selection of the references and the order of the executions depend on **Con**. Name M_{inp} the input map of the Unit creation formula call. The first reference execution has input map M_{inp} while each other reference execution has input map the previous execution's output map. The output map of the Unit creation formula emerge as the result of imprinting to M_{inp} all the Unit's child-objects excluding terminal Primitives whose property list dissatisfies **FLT**. All the reference executions have **U** as the target Unit.

G-CAD provides the connectives **G_AND**, **G_OR**, **G_ONE**, **G_ALL** and **G_REC** presented here.

a) The connective "**G_AND**"

All the formula references appearing in the reference section of the Unit Definition are executed in appearance order. The Unit creation formula is confirmed if and only if all the reference executions are confirmed. The output values of the Unit creation formula are the values assigned to the symbols in the output section of the Unit Definition.

Syntactical constraints:

Each symbol in the output section of a formula reference should not be included in:

- the input section of the Unit Definition
- the input section of the formula reference
- either the input or the output section of a previous reference within the Unit Definition

b) The connective "**G_OR**"

A formula reference randomly selected from the reference section of the Unit Definition is executed. The Unit creation formula is confirmed if and only if the reference's execution is confirmed. The output values of the Unit creation formula are the values assigned to the symbols in the output section of the Unit Definition.

Syntactical constraints:

- Each symbol in the input section of a formula reference should be included either in the input section of the Unit Definition or in the property section as a property name.
- Each symbol in the output section of a formula reference should not be included in:
  - the input section of the Unit Definition
  - the input section of the formula reference
- Each symbol in the output section of the Unit Definition should be included in the output section of each formula reference within the reference section of the Unit Definition.

c) The connective "**G_ONE**" (existential quantifier)

The reference section of the Unit Definition includes only one formula reference. The last symbol in the input section of this reference is assigned a list, name it **L**. The reference is executed once having the last input term substituted by a randomly selected element of **L**. The Unit creation formula is confirmed if and only if the reference's execution is confirmed. The output values of the Unit creation formula are the values assigned to the symbols in the output section of the Unit Definition.

Syntactical constraints:

Each symbol in the output section of the formula reference should not be included in:

- the input section of the Unit Definition
- the input section of the formula reference

d) The connective "**G_ALL**" (universal quantifier)

The reference section of the Unit Definition includes only one formula reference. The last symbol in the input section of this reference is assigned a list, name it **L**. The reference is executed once for each element of **L**. In each execution the specific element substitutes the last input term of the reference. The Unit creation formula is confirmed if and only if all the executions are confirmed.

The procedure defining the output values of the Unit creation formula is presented here:

- Let $\mathbf{L_1}$, … $\mathbf{L_k}$ be empty lists where **k** is the size of the output section of the reference.
- For each execution append the $\mathbf{i^{th}}$ (i = 1, … k) output value of the execution to the list $\mathbf{L_i}$ as the last element.
- Assign $\mathbf{L_i}$ to the $\mathbf{i^{th}}$ symbol in the output section of the reference.

The output values of the Unit creation formula are the values assigned to the symbols in the output section of the Unit Definition.

Syntactical constraints:

Each symbol in the output section of the formula reference should not be included in:

- the input section of the Unit Definition
- the input section of the formula reference

e) The connective "**G_REC**" (recursion)

The Unit Definition includes two formula references. The first one refers to a secondary formula, which represents the recursion termination condition, while the second one refers to a formula of any type. The Unit's construction process starts with the execution of the first reference. If the execution is confirmed then the process ends while the Unit creation formula is confirmed. Otherwise the second reference is executed. In the case of confirmation the Unit's construction process is repeated, whereas in the case of disconfirmation the process ends while the Unit creation formula is disconfirmed. Any symbol appearing both in the input section of the first reference and in the output section of the second reference represents a recursion term (i.e. a term that changes in each recursion). The output values of the Unit creation formula are the values assigned to the symbols in the output section of the Unit Definition.

Syntactical constraints:

- Each symbol in the input section of a formula reference should be included either in the input section of the Unit Definition or in the property section as a property name.
- Each symbol in the output section of the first formula reference should not be included in:
  - the input section of the Unit Definition
  - the input section of the formula reference
- At least one symbol in the output section of the second formula reference should be included in the input section of the Unit Definition
- Each symbol in the output section of the Unit Definition should be included in the output section of the second formula reference.

2.4 *Random decisions to be evolved within G-CAD formulae*

The combination of all the random decisions made during the execution of a G-CAD program is to be evolved by the computational system within GENETICA's environment. The random decisions programmed within the G-CAD formulae are summarized here:

- The argument of both Primitive and Instance creation formulae (see § 2.2.a, 2.2.c) includes a Positioning List. Positioning Lists (see § 1.9) do not necessarily specify a single positioning of a design object: different positions, orientations and mirroring conditions of the design object may be defined by the Positioning List's specifications. In this case the positioning depends on random decisions.
- The argument of a Primitive creation formula (see § 2.2.a) includes a Sizing List having Dimension Definition Formulae as elements. The Dimension Definition Formulae **plc_A_in_G** and **plc_dd_in_G** (see Appendix A.1) include random decisions affecting the dimensions of the Primitive to be created.
- When a Unit creation formula constructed with the connective **G_OR** (see § 2.3.b) is called the reference to be executed is selected at random.
- When a Unit creation formula constructed with the connective **G_ONE** (see § 2.3.c) is called a random selection determines the last input term of the referenced formula to be called.
- The secondary formulae **G_MEM** and **G_IN** (see Appendix A.2) include random decisions

3 Data organization for design problem solving in G-CAD

3.1 *The problem formulation*

The problem formulation, which is an input vector value for the GENETICA program that implements G-CAD, includes the following parts:

a) Standard (non evolvable) Units, which constitute components of a solution.
b) Units to be evolved. Each evolvable Unit is represented by the name of a Unit creation formula, the formula's input map and a vector value for the formula's input section.
c) A G-CAD program including the Unit Definitions for the evolvable Units.
d) The G-CAD constants.

Specifically, the problem formulation is represented as a list **(W, Q, P, c_1, … c_n)** where:

- **W** is the World Unit (see § 1.8) that includes the non evolvable Units as sub-objects. A World Unit without sub-objects is also legal.
- **Q** is a list of quadruples (four element lists), each one representing a Unit to be evolved. Each quadruple has the form **(G, L, N, V)** where:
  - **G** and **L** are lists defining the input map for the Unit creation formula. **G** has the form **(R, D_1, D_2)** where **R** is the list of the rows of the input map while D_1 and D_2 are the dimension lists (see § 1.3). Rows are lists of equal size, name it **s**, having integers as elements. The **i^{th}** integer (i = 1, … s) corresponds to the **i^{th}** cell of the row. **L** is a list of property lists. The integer corresponding to each cell of the map represents the order of an element of **L**. The element is assigned to the cell.
  - **N** is the Unit creation formula's name viewed as a string. The formula's target Unit is the World Unit.

- **V** is a list of values for the input section of the Unit named **N**. Each symbol in the input section is assigned the homologous value in **V**.

Evolving Units are created in the order they appear in **Q**. When an evolving Unit is created it is appended at the World Unit as the last element. The last output term of the Unit creation formula referenced in the last quadruple in **Q** should be a real. This term represents the magnitude to be maximized in the case of an optimization problem.

- **P** is a G-CAD program as a list of Unit Definitions. Each Unit creation formula referenced in **Q** should be included in **P**.
- $\mathbf{c_1, \ldots c_n}$ are standard values representing the G-CAD constants.

3.2 *The representation of a solution*

The output vector value for the GENETICA program that implements G-CAD is a two-element list. The first element is the World Unit where the evolved Units represented in the list **Q** (introduced in the previous paragraph) have been appended. The second element is a list having the form $\mathbf{(L_1, \ldots L_n)}$ where **n** is the size of **Q** and $\mathbf{L_i}$ (i = 1, … n) is the output vector value of the Unit creation formula referenced in the $\mathbf{i^{th}}$ element of **Q**.

Let $\mathbf{q_n = (G_n, L_n, N_n, V_n)}$ be the last quadruple in **Q**. The evolved Unit, name it $\mathbf{U_n}$, represented by $\mathbf{q_n}$ is considered as the solution of the design problem.

A CAD interface presents the solution as an architectural drawing with respect to user-defined specifications. Specific properties of Primitives within the standard and the evolving Units link each Unit with either a fixed or a parametric drawing. The CAD interface:

- Draws the drawing linked to each sub-Unit of $\mathbf{U_n}$ with respect to the sub-Unit's positioning on the Cartesian Plane
- Creates the Spatial Map, name it $\mathbf{M_{out}}$, occurring by imprinting $\mathbf{U_n}$ at the Spatial Map defined by $\mathbf{G_n}$ and $\mathbf{L_n}$.
- Draws different types of walls at the boundaries between regions of $\mathbf{M_{out}}$ specified by user-defined Queries.

APPENDIX A

A.1 Dimension Definition Formulae

A Dimension Definition Formula is the formula—referenced in the Sizing List (see § 1.10, 2.2.a) within the argument of a Primitive creation formula—defines a dimension of the Primitive (viewed as a rectangle) given the Primitive's OCS, and the Cartesian Plane's region, name it **R**, where the Primitive should be included. Both the Primitive's OCS and **R** are defined by the Primitive's creation formula call.

Within the following presentation the term **D** denotes the dimension to be defined, while the term $\mathbf{D_R}$ denotes the maximum allowed **D** value specified both by the Primitive's OCS and the region **R**. **D** is explicitly defined if it appears in the argument of a Dimension Definition Formula.

The Dimension Definition Formulae are presented here in the format

<formula_name, input_vector_value>:

"**plc_ddx_std**" **(D, $\mathbf{D_{max}}$)**

Defines the Primitive's dimension towards either the *X* or the *Y* axis of the Primitive's OCS. It is confirmed if and only if $\mathbf{0 < D \leq \min \{D_R, D_{max}\}}$.

"**plc_ddx_max**" **($\mathbf{D_{min}}$, $\mathbf{D_{max}}$)**

Defines the Primitive's length towards either the *X* or the *Y* axis of the Primitive's OCS. It is confirmed if and only if $\mathbf{0 < D_{min} \leq \min \{D_R, D_{max}\}}$. In the case of confirmation $\mathbf{D = \min \{D_R, D_{max}\}}$.

"**plc_A_std**" **(A, $\mathbf{D_{max}}$)**, where **A** represents the area of the Primitive.

Defines the Primitive's length only towards the *Y* axis of the Primitive's OCS. As a consequence appears only as the second Dimension Definition Formula within a Sizing List. It is confirmed if and only if $\mathbf{0 < \frac{A}{D_X} \leq \min \{D_R, D_{max}\}}$ where $\mathbf{D_X}$ is the length of the Primitive towards the *X* axis of the Primitive's OCS, which has been defined by the first Dimension Definition Formula of the Sizing List. In the case of confirmation $\mathbf{D = \frac{A}{D_X}}$.

"**plc_A_in_G**" **(("ITE_call_BC"), ("plc_get_G1"), ("plc_get_G2"), ("in"), $\mathbf{A_{min}}$, $\mathbf{A_{max}}$, $\mathbf{D_{Ymin}}$, $\mathbf{D_{Ymax}}$)**,

where $\mathbf{A_{min}}$ and $\mathbf{A_{max}}$ represent the minimum and the maximum acceptable area of the Primitive respectively.

Defines the Primitive's length only towards the *Y* axis of the Primitive's OCS. As a consequence appears only as the second Dimension Definition Formula within a Sizing List. Let $\mathbf{D_X}$ be the length of the Primitive towards the *X* axis of the Primitive's OCS, which has been defined by the first Dimension Definition Formula within the Primitive's Sizing List. Define:

$$\mathbf{D_{min} = max\ \{\frac{A_{min}}{D_X}, D_{Ymin}\}}, \qquad \mathbf{D_{max} = min\ \{\frac{A_{max}}{D_X}, D_{Ymax}, D_R\}}.$$

plc_A_in_G is confirmed if and only if $\mathbf{D_{min} \le D_{max}}$. Let $\mathbf{SM_R}$ be the input map of the call to the Primitive's creation formula, viewed from the Primitive's OCS. If there exist *Y* coordinates of row boundaries of $\mathbf{SM_R}$ within the interval $\mathbf{[D_{min}, D_{max}]}$ then the value of one of these coordinates, selected randomly, is assigned to **D**: this makes certain that the Primitive fits in existing rows, so no new rows will be created as a result of the Primitive's imprinting. Otherwise **D** is assigned a random real value in the interval $\mathbf{[D_{min}, D_{max}]}$.

"**plc_dd_in_G**" **(("ITE_call_BC"), ("plc_get_G1"), ("plc_get_G2"), ("in"),** $\mathbf{D_{min}}$**,** $\mathbf{D_{max}}$**)**

Defines the Primitive's length towards either the *X* or the *Y* axis of the Primitive's OCS. It is confirmed if and only if $\mathbf{D_{min} \le min\ \{D_{max}, D_R\}}$. Let $\mathbf{SM_R}$ be the input map of the call to the Primitive's creation formula, viewed from the Primitive's OCS. If there exist coordinates of column (respectively row) boundaries of $\mathbf{SM_R}$ within the interval $\mathbf{[D_{min}, min\ \{D_{max}, D_R\}]}$ on the *X* (respectively *Y*) axis, then the value of one of these coordinates, selected randomly, is assigned to **D**: this makes certain that the Primitive fits in existing columns (respectively rows), so no new columns (respectively rows) will be created as a result of the Primitive's imprinting. Otherwise **D** is assigned a random real value in the interval $\mathbf{[D_{min}, D_{max}]}$.

A.2 Secondary Formulae

The secondary G-CAD formulae, i.e. the G-CAD formulae not constructing design objects, are presented here as triplets having the form $\mathbf{<F_N, V_{inp}, V_{out}>}$ where $\mathbf{F_N}$ is the formula's name viewed as a string, while $\mathbf{V_{inp}}$ and $\mathbf{V_{out}}$ are the lists of the input and the output terms respectively. The string **str**, appearing as the first input term of each secondary formula, takes either the value "**call**" or the value "**at**". The former indicates a normal call to the formula while the latter an "always true" call (see GENETICA's documentation: *GENETICA_Documentation.PDF* : § 3.2.1.d).

"**G_ADD**" **(str,** $\mathbf{L_{in}}$**, V)** $\mathbf{(L_{out})}$

which is always confirmed. If $\mathbf{L_{in}}$ is a list and **V** is a term of any type then $\mathbf{L_{out}}$ is the list produced by appending **V** as an element at the end of $\mathbf{L_{in}}$.

"**G_ADDPR**" **(str, N, V) ( )**

which is always confirmed.

N is a term of arbitrary type, which represents a property name.
V is a term of arbitrary type, which represents a property value.

The pair **(N, V)** is appended as an element to the property list of the target Unit of the Unit creation formula that calls "**G_ADDPR**".

"**G_AMUCS**" **(str, Pos_{in}, UCS, (1, 0, -1, 0), 1.0, 4.0) (Pos_{out})**

which is confirmed if and only if **Pos_{in}** is a Positioning List and **UCS** is a OCS. In the case of confirmation **Pos_{out}** is the Positioning List resulting by transforming **Pos_{in}** from **UCS** to the World coordinate system (see § 1.6). The transformation affects only the third and the fourth element of **Pos_{in}** i.e. the direction vectors and the mirroring coefficients (see § 1.9).

"**G_AREA**" **(str, QR, 0, 1, 'i', 'l') (A)**

which is always confirmed. **QR** is a Query which is applied to the input map. **A** is the area of the map's region that satisfies **QR**.

"**G_BF**" **(str, L_{in}) (L_{out})**

which is confirmed if and only if **L_{in}** is a non empty list. In the case of confirmation **L_{out}** is the remaining part of **L_{in}** after removing the first element.

"**G_BL**" **(str, L_{in}) (L_{out})**

which is confirmed if and only if **L_{in}** is a non empty list. In the case of confirmation **L_{out}** is the remaining part of **L_{in}** after removing the last element.

"**G_DVS**" **(str, V_1, V_2) (V)**

which is confirmed if and only if **V_1** and **V_2** are reals. In the case of confirmation **V** is the quotient $\frac{V_1}{V_2}$.

"**G_EMPTY**" **(str, L_{in}) (L_{out})**

which is confirmed if and only if **L_{in}** is an empty list. **L_{out}** is the empty list.

"**G_FI**" **(str, L) (V)**

which is confirmed if and only if **L** is a non empty list. In the case of confirmation **V** is the first element of **L**.

"**G_IN**" **(str, V_1, V_2, Q) (V)**

which is confirmed if and only if **V_1**, **V_2** and **Q** are reals, where **$V_1 < V_2$** and **$Q > 0$**. In the case of confirmation **G_IN** generates a random real, name it **r**, within the interval **$[V_1, V_2]$**, calculates the integer part, name it **i**, of the quotient $\frac{r}{Q}$ and defines **V** to be the product **i·Q**.

"**G_IUPR**" **(str, p, N) (V)**

which is confirmed if and only if **p** is a pointer to a Unit, name it **U**, where **U** constitutes sub-Unit of the World Unit, while the property list of **U** includes a property named **N**. In the case of confirmation **V** is the value of the property named **N**.

"**G_JO**" **(str, L_1, L_2)(L_{out})**

which is always confirmed. If both L_1 and L_2 are lists then L_{out} is the list produced by adjoining L_2 to L_1. Specifically, if $L_1 = (x_1, \ldots x_n)$ and $L_2 = (y_1, \ldots y_k)$ then $L_{out} = (x_1, \ldots x_n, y_1, \ldots y_k)$.

"**G_LA**" **(str, L) (V)**

which is confirmed if and only if **L** is a non empty list. In the case of confirmation **V** is the last element of **L**.

"**G_MEM**" **(str, L) (V)**

which is confirmed if and only if **L** is a non empty list. In the case of confirmation **V** is an randomly selected element of **L**.

"**G_MLT**" **(str, V_1, V_2) (V)**

which is confirmed if and only if V_1 and V_2 are reals. In the case of confirmation **V** is the product $V_1 \cdot V_2$.

"**G_MNS**" **(str, V_1, V_2) (V)**

which is confirmed if and only if V_1 and V_2 are reals. In the case of confirmation **V** is the difference $V_1 - V_2$.

"**G_PL**" **(str, PL, N) (V)**

which is confirmed if and only if **PL** is a property list and **N** is a property name in **PL**. In the case of confirmation **V** is the value of the property named **N** in **PL**.

"**G_PLS**" **(str, V_1, V_2) (V)**

which is confirmed if and only if V_1 and V_2 are reals. In the case of confirmation **V** is the sum $V_1 + V_2$.

"**G_SBS**" **(str, L_{in}, t_{old}, t_{new}) (L_{out})**

which is always confirmed.

L_{in}	is a list
t_{old}	is a term of any type except list
t_{new}	is a term of any type except list
L_{out}	is the list derived from L_{in} if each terminal (atom) in L_{in} equal to e_{old} is substituted by e_{new}.

APPENDIX B

A G-CAD application: design of a hotel floorplan

A small hotel's floorplan, having eight identical apartments, a corridor and a stairwell is to be designed. An apartment consists of a main room and a WC, while the main room should be adjacent to a corridor. The dimensions of the apartment's rooms should satisfy specific geometric constraints. The apartments can be combined in specific ways, while no apartment-to-apartment overlapping is allowed. The stairwell should be adjacent to the corridor without overlapping an apartment. Specific objects representing openings, furniture or bathroom fittings should be placed in an apartment. These objects should satisfy both object-to-object and object-to-apartment geometric relations. The area of the built space, i.e. the space occupied by either the apartments or the corridor, should be minimized.

The formal problem statement (see § 3.1), including the definition of the initial condition of the World Unit, the definitions of the evolvable Units, and a G-CAD program as a list of Unit Definitions, is presented in the following paragraphs. Wherever G-CAD code appears within this presentation, it is directly copied from the application run in the computer. This makes certain that neither errors nor ambiguities appear in the code. Within the G-CAD code, list elements are separated by spaces (without commas) while comments start with a semicolon (;).

B.1 The World Unit

The World Unit, which constitutes the first element of the problem formulation, includes the standard child-Units presented in Figure 3 as assemblies of Primitives.

Consider each Primitive within a standard Unit having a name and a geometric definition, the latter represented by the Primitive's OCS and dimensions (see § 1.7). Then the standard Unit can be defined by a list of three elements:

1. The Unit's property list (see § 1.8: $\mathbf{PL_U}$)
2. A list of pairs, each one representing a Primitive of the Unit, where the first element of each pair is the Primitive's name while the second element is the Primitive's property list (see § 1.7: **PL**)
3. The name of a specific Primitive, referred to as the Unit's "first Primitive", whose OCS is considered as the Unit's OCS.

This list will be referred to as the RUL list of the Unit, where RUL stands for Registered Unit List.

Within a Unit's RUL list:

- The Unit's property named "**ID**", which stands for "Identity", allows reference to the Unit so that the Unit can be instanced.
- A Primitive's property named "**BLOCK**" is assigned the name of a drawing associated to the Primitive. This drawing constitutes the graphical representation of the Unit.
- A Primitive's property named "**ATTR**" is assigned a list of spatial attributes which are to be assigned to a Spatial Map's cells. The names of these attributes are given the following interpretations:
  - "**DS**" : Door Space (the space occupied by a door)
  - "**FS**" : Free Space (space that should not be occupied by physical objects)
  - "**OS**" : Object Space (space occupied by a physical object)
  - "**OPN**" : Opening (space adjacent to an opening)

- "**WS**" : Window Space (room space adjacent to the window).

The RUL lists of the standard Units of the hotel application are presented here, under the Unit names appearing in Figure 3.

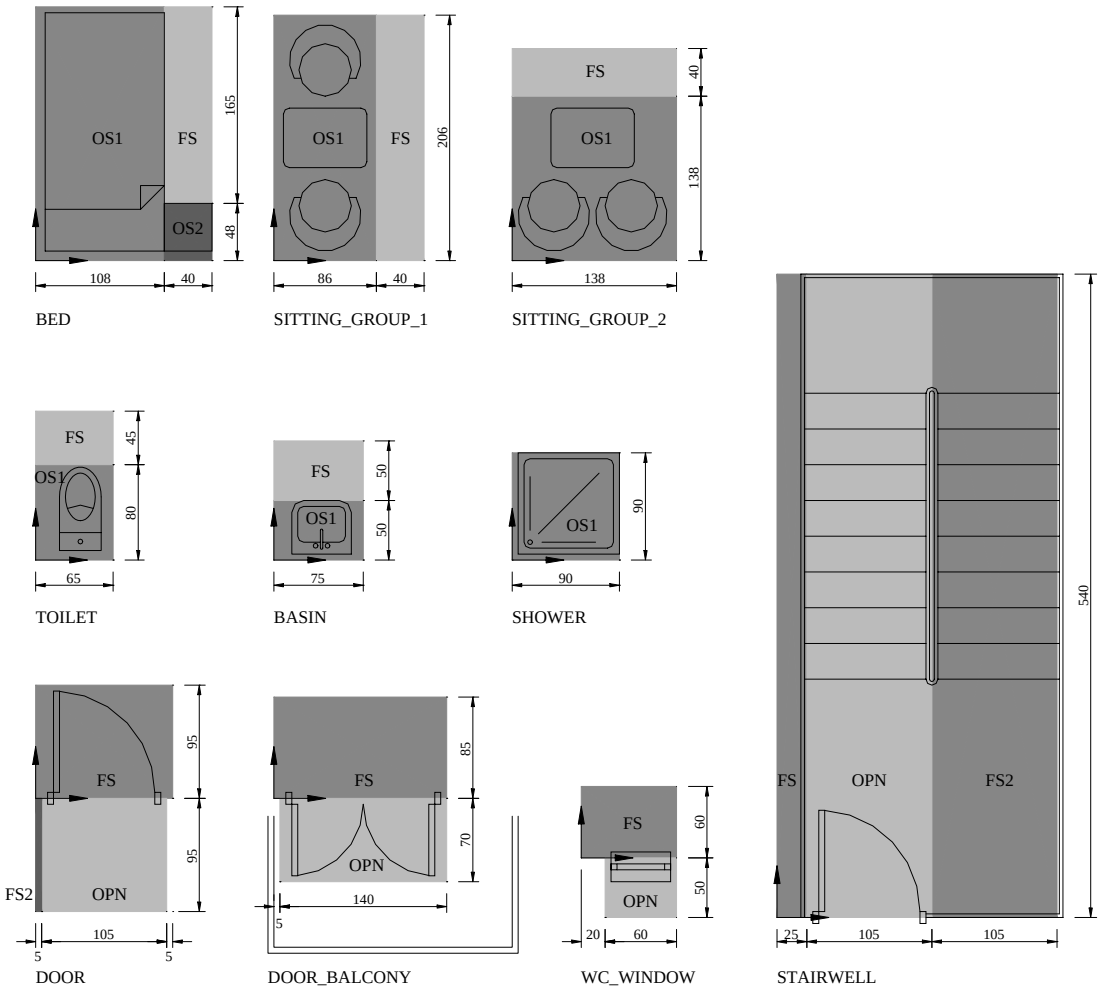


Figure 3

The standard (non evolvable) units within the World Unit of the hotel application. Each standard Unit includes Primitives represented as shaded rectangles. Each Primitive is characterized by a name which appears in the Unit's RUL list. The dimensions of the Primitives represent centimeters. The OCS of each Unit appears as a system of two vectors, perpendicular to each other, where the horizontal vector represents the X axis of the OCS. A specific property of each Unit refers to a drawing that represents the Unit as a physical object.

DOOR

```
 (
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "DOOR1" )
    )
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "FS"
            (
                (   "BLOCK"     "DOOR1" )
                (   "ATTR"      "DS"    )
                (   "APA"
                    (   "log_not"
                        (   "log_eq"    (   "NONE"      1   )   )
                    )
                )
            )
        )
        (   "FS2"
            (
                (   "ATTR"      "FS"    )
                (   "APA"
                    (   "log_not"
                        (   "log_eq"    (   "NONE"      1   )   )
                    )
                )
            )
        )
        (   "OPN"
            (
                (   "ATTR"      "OPN"   )
                (   "APA"
                    (   "log_not"
                        (   "log_eq"    (   "NONE"      1   )   )
                    )
                )
            )
        )

    )
    "FS"                                ; ------------------ FIRST PRIMITIVE
)
```

DOOR-BALCONY

```
 (
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "DOOR2" )
    )
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "FS"
            (
                (   "BLOCK"     "DOOR2" )
                (   "ATTR"      "DS"    )
                (   "APA"
                    (   "log_not"
                        (   "log_eq"    (   "NONE"      1   )   )
                    )
                )
            )
        )
```

```
        (   "OPN"
            (
                (   "ATTR"      "OPN"   )
                (   "APA"
                    (   "log_not"
                        (   "log_eq"    (   "NONE"      1   )   )
                    )
                )
            )
        )

    )
    "FS"                            ; ------------------ FIRST PRIMITIVE
)
```

SITTING GROUP 1

```
(
    (                               ; ------------------ UNIT PROPERTIES
        (   "ID"    "SIT1"  )
    )
    (                               ; ------------------ PRIMITIVES' PROP. LISTS
        (   "OS1"
            (
                (   "BLOCK"     "SIT1"  )
                (   "ATTR"      "OS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                            (   "log_eq"    (   "ATTR"      ()      )   )
                        )
                    )
                )
            )
        )
        (   "FS"
            (
                (   "ATTR"      "FS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"      "OS"    )   )
                            )
                        )
                    )
                )
            )
        )
    )
    "OS1"                           ; ------------------ FIRST PRIMITIVE
)
```

SITTING GROUP 2

```
(
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "SIT3"  )
    )
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "OS1"
            (
                (   "BLOCK"     "SIT3"  )
                (   "ATTR"      "OS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                            (   "log_eq"    (   "ATTR"      ()      )   )
                        )
                    )
                )
            )
        )
        (   "FS"
            (
                (   "ATTR"      "FS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"      "OS"    )   )
                            )
                        )
                    )
                )
            )
        )

    )
    "OS1"                               ; ------------------ FIRST PRIMITIVE
)
```

BED

```
(
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "BED"   )
    )
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "OS1"
            (
                (   "BLOCK"     "BED"   )
                (   "ATTR"      "OS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                            (   "log_eq"    (   "ATTR"          ()  )   )
                        )
                    )
                )
            )
        )
```

```
        (   "OS2"
            (
                (   "ATTR"      "OS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                            (   "log_eq"    (   "ATTR"      ()      )   )
                        )
                    )
                )
            )
        )
        (   "FS"
            (
                (   "ATTR"      "FS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"      "OS"    )   )
                            )
                        )
                    )
                )
            )
        )

    )
    "OS1"                               ; ------------------ FIRST PRIMITIVE
)
```

WC WINDOW

```
(
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "WWC"   )
    )
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "FS"
            (
                (   "ATTR"      "WS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "WC"    )   )
                            (   "log_eq"    (   "ATTR"      ()      )   )
                        )
                    )
                )
            )
        )
        (   "OPN"
            (
                (   "BLOCK"     "WWC"   )
                (   "ATTR"      "OPN"   )
                (   "APA"
                    (   "log_not"
                        (   "log_ism"   (   "BUILT"     1   )   )
                    )
                )
            )
        )
    )
    "FS"                                ; ------------------ FIRST PRIMITIVE
)
```

BASIN

```
 (
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "BASIN" )
    )
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "OS1"
            (
                (   "BLOCK"     "BASIN" )
                (   "ATTR"      "OS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "WC"    )   )
                            (   "log_eq"    (   "ATTR"      ()      )   )
                        )
                    )
                )
            )
        )
        (   "FS"
            (
                (   "ATTR"      "FS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "WC"    )   )
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"      "OS"    )   )
                            )
                        )
                    )
                )
            )
        )

    )
    "OS1"                               ; ------------------ FIRST PRIMITIVE
)
```

SHOWER

```
(
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "DOUCHE"    )
    )
```

```
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "OS1"
            (
                (   "BLOCK"     "DOUCHE"    )
                (   "ATTR"      "OS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "WC"    )   )
                            (   "log_not"
                                (   "log_or"
                                    (
                                        (   "log_ism"   (   "ATTR"   "OS"  ) )
                                        (   "log_ism"   (   "ATTR"   "FS"  ) )
                                        (   "log_ism"   (   "ATTR"   "DS"  ) )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
    )

    "OS1"                               ; ------------------ FIRST PRIMITIVE
)
```

TOILET

```
(
    (                                   ; ------------------ UNIT PROPERTIES
        (   "ID"    "TOIL"  )
    )
    (                                   ; ------------------ PRIMITIVES' PROP. LISTS
        (   "OS1"
            (
                (   "BLOCK"     "TOILETTE"  )
                (   "ATTR"      "OS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "WC"    )   )
                            (   "log_not"
                                (   "log_or"
                                    (
                                        (   "log_ism"   (   "ATTR"   "OS" ) )
                                        (   "log_ism"   (   "ATTR"   "FS" ) )
                                        (   "log_ism"   (   "ATTR"   "DS" ) )
                                    )
                                )
                            )
                        )
                    )
                )
            )
        )
```

```
        (   "FS"
            (
                (   "ATTR"      "FS"    )
                (   "APA"
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "WC"    )   )
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"      "OS"    )   )
                            )
                        )
                    )
                )
            )
        )

    )

    "OS1"                           ; ------------------ FIRST PRIMITIVE
)
```

STAIRWELL

```
(
    (                               ; ------------------ UNIT PROPERTIES
        (   "ID"    "STAIR" )
    )
    (                               ; ------------------ PRIMITIVES' PROP. LISTS
        (   "FS"
            (
                (   "BLOCK"     "STAIR" )
                (   "APA"
                    (   "log_not"
                        (   "log_ism"   (   "BUILT"     1   )   )
                    )
                )
            )
        )
        (   "OPN"
            (
                (   "ATTR"      "OPN"   )
                (   "APA"
                    (   "log_not"
                        (   "log_ism"   (   "BUILT"     1   )   )
                    )
                )
            )
        )
        (   "FS2"
            (
                (   "APA"
                    (   "log_not"
                        (   "log_ism"   (   "BUILT"     1   )   )
                    )
                )
            )
        )
    )

    "FS"                            ; ------------------ FIRST PRIMITIVE
)
```

B.2 Evolvable Units

The representation of an evolvable Unit, within the problem formulation, includes the input map, the formula name and an input vector value of a Unit creation formula (see § 3.1). The hotel application includes three evolvable Units. The list of the evolvable Units, which comes after the World Unit in the problem formulation list has the form **((G, L, "TYPE", V), (G, L, "TYPE_R", ( )), (G, L, "TEST_P", ( ))),** where:

- the lists **G** and **L** define a Spatial Map having nine cells which form three rows and three columns. All the cells are assigned the property list **(("BUILT", ( )), ("ATTR", ( )), ("IID", ( )))**, where the properties **BUILT**, **ATTR** and **IID** respectively accept values:
  - a list indicating either built space if includes the value **1** or not built space otherwise
  - a list of arbitrary spatial attributes
  - a list where the identities of the imprinted Instances are to be registered. The value of a specific property named **IID**, which stands for "Instance's Identity", within an Instance's "Instance List" (see § 2.2.c) is considered as the Instance's identity.

  The pair **("BASE", 1)** is appended as an element to the property list of the Map's central cell. Due to this property, the central cell can be referenced by the Positioning List of the formula creating the first object of each evolvable Unit.
- The Unit creation formulae **TYPE**, **TYPE_R** and **TEST_P** create Units that respectively represent:
  - as an assembly of two Primitives, each one representing a room (Figure 4)
  - an apartment type equipped with openings, furniture and bathroom fittings
  - a hotel floorplan as a configuration of eight apartments sharing a corridor and a stairwell

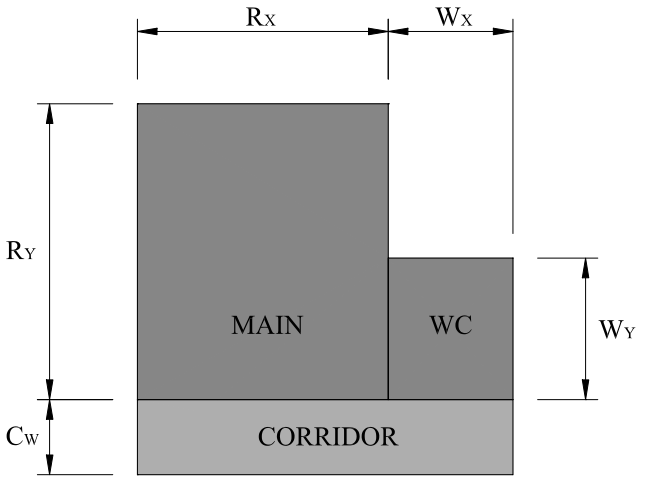


Figure 4

*The Primitives named "**MAIN**" and "**WC**" compose a **TYPE** Unit. The Primitive named "**MAIN**" represents the main room while the Primitive named "**WC**" represents the **WC** of the apartment. The width of the Primitive named "**CORRIDOR**", which represents a fragment of a corridor connecting different apartments, affects the definition of the dimensions of the **TYPE** Primitives.*

- **V** is an input vector value for the **TYPE** formula, while the formulae **TYPE_R** and **TEST_P** have no input values.

The input terms for the **TYPE** formula are presented here with respect to their order in the formula's input section:

RX_MIN	which is the minimum acceptable $\mathbf{R_X}$ dimension for the "**MAIN**" Primitive (Figure 4)	(real)
RX_MAX	which is the maximum acceptable $\mathbf{R_X}$ dimension for the "**MAIN**" Primitive	(real)
CWIDTH	which is the width $\mathbf{C_W}$ of a corridor adjacent to the apartment (Figure 4)	(real)
WX_MIN	which is the minimum acceptable $\mathbf{W_X}$ dimension for the "**WC**" Primitive (Figure 4)	(real)
WX_MAX	which is the maximum acceptable $\mathbf{W_X}$ dimension for the "**WC**" Primitive	(real)
WE_MIN	which is the minimum acceptable area for the "**WC**" Primitive	(real)
WE_MAX	which is the maximum acceptable area for the "**WC**" Primitive	(real)
PL_MAIN	which is the property list of the "**MAIN**" Primitive	
PLC_MAIN	which is a Positioning List for the "**MAIN**" Primitive	
PL_WC	which is the property list of the "**WC**" Primitive	
PLC_WC	which is a Positioning List for the "**WC**" Primitive	

The value assignment for the input terms of the **TYPE** formula is presented here:

```
RX_MIN      350.0
RX_MAX      390.0
CWIDTH      130.0
WX_MIN      170.0
WX_MAX      230.0
WE_MIN      35000.0
WE_MAX      40000.0
PL_MAIN     (
                (   "BUILT"     1       )
                (   "ROOM"      "MAIN"  )
                (   "APA"
                    (   "log_not"   (   "log_ism"   (   "BUILT"     1   )   )   )
                )
            )
PLC_MAIN    (
                (   "log_not"
                    (   "log_eq"    (   "BASE"      1   )   )
                )
                (   "log_eq"    (   "BASE"      1   )   )
                (   1.0 )
                (   1.0 )
                50.0
            )
PL_WC       (
                (   "BUILT"     1       )
                (   "ROOM"      "WC"    )
                (   "APA"
                    (   "log_not"   (   "log_ism"   (   "BUILT"     1   )   )   )
                )
            )
PLC_WC      (
                (   "log_not"
                    (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                )
                (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                (   4.0     )
                (   -1.0    )
                100.0
            )
```

B.3 The G-CAD program

B.3.1 *A G-CAD formula specifying the dimensions of an apartment*

The G-CAD formula **G_HTYPE1** defines the Sizing Lists of the two Primitives that compose the **TYPE** Unit (Figure 4). The definition respects specific geometric constraints. Although **G_HTYPE1** can be constructed in G-CAD, it has been developed in GENETICA, for the sake of simplicity and computational efficiency. The definition of **G_HTYPE1** is presented here with respect to the notation used in the Appendix A.2.

G_HTYPE1 (str, DDF, R_{Xmin}, R_{Xmax}, C_{width}, W_{Xmin}, W_{Xmax}, W_{Amin}, W_{Amax}) (DIM_R, DIM_W)

where:

- **DDF = "plc_ddx_std"**,
- The terms R_{Xmin}, R_{Xmax}, C_{width}, W_{Xmin}, W_{Xmax}, W_{Amin} and W_{Amax} respectively represent the terms **RX_MIN**, **RX_MAX**, **CWIDTH**, **WX_MIN**, **WX_MAX**, **WE_MIN** and **WE_MAX** defined in § B.2
- DIM_R is a Sizing List for the "MAIN" Primitive
- DIM_W is a Sizing List for the "WC" Primitive

The Sizing List DIM_R defines the dimensions R_X and R_Y, while the Sizing List DIM_W defines the dimensions W_X and W_Y appearing in Figure 4. The dimensions respect the following geometric constraints:

$$R_{Xmin} < R_X < R_{Xmax}$$

$$W_X = W_{Xmax} - (R_X - R_{Xmin}) \cdot \frac{W_{Xmax} - W_{Xmin}}{R_{Xmax} - R_{Xmin}}$$

$$R_Y = R_X + W_X - C_{width}$$

$$W_{Amin} < W_X \cdot W_Y < W_{Amax}$$

B.3.2 *The Unit Definitions*

All the Unit Definitions within the G-CAD program are presented here. A short description of the functionality of each Unit Definition precedes each Unit Definition's code. Within these descriptions, a Unit or an Instance is sometimes referred to by the name of the Unit Definition that creates the Unit or the instanced Unit respectively, e.g. "the Unit **TYPE**" means "the Unit created by the Unit Definition **TYPE**", while "the Instance **TYPE**" means "the Instance of the Unit created by the Unit Definition **TYPE**".

The Unit **TEST_P** represents a hotel's floorplan consisting of eight identical apartments sharing a corridor and a stairway. The construction process, formally defined in the reference section, is presented here:

- Create a **TYPE** Instance (see § 2.2.c) adjacent to the central cell of the input map. The Unit **TYPE** (see the Unit Definition **TYPE**, later in the code) has a property named "**ID**", which stands for "identity", assigned the value "**T1**"; as a consequence it is identified by the Query **("log_eq", ("ID", "T1"))** represented by the property **IUS_T** defined in the property section of the Unit Definition **TEST_P**. The Positioning List **PLC_I1**, also defined in the property section, defines a single placement of the Instance at the top of the input map's region that has the property "**BASE**" assigned the value **1**. The Instance List (see § 2.2.c), represented by the property **CPL_I1**, appends the value "**I1**" to the **IID** list (see Appendix B.2) of the output map's cells occupied by the Instance.
- Call the secondary formula **G_IUPR** (see Appendix A.1) to get the value of the property **LEN** of the instanced Unit. **LEN** specifies the length of the apartment type, i.e. the sum of the dimensions $\mathbf{R_X}$ and $\mathbf{W_X}$ appearing in Figure 4. Name **C_LEN_1** the length. Then calculate **C_LEN_2** as the double length.
- Perform successive calls to the Unit Definition **PLACE** to place the remaining seven Instances of the Unit **TYPE**, each one adjacent to the previous Instance. The Unit **PLACE** (see **PLACE** Unit Definition) specifies the positioning method—by selecting a method compatible with the positioning of the previous Instances—and realizes the positioning of the new Instance.
- Create an Instance of the "**STAIRWELL**" standard Unit (see Figure 3) in the non-built space, adjacent to space having the spatial attribute "**COR**" which denotes a corridor. The Unit is specified by the Query **IUS_STR**, since its **ID** property is assigned the value "**STAIR**", and it is placed by the Positioning List **PLC_STR**.
- Call the secondary formula **G_AREA** to calculate the area of the built space; the built space is specified by the Query **A_CL** which is defined in the property section of the Unit Definition **TEST_P**.
- Divide the real **50000000.0** (property **C_MIL**) by the area of the built space and return the result as the magnitude to be maximized.

```
(   "TEST_P"                        ; --------------   UNIT TEST_P
    (   "G_AND"
        ()                                          ;   TEST_P INPUT SECTION
        (                                           ;   TEST_P PROP. SECTION
            (   "ID"        "TM1"   )
            (   "FSTR"      "at"    )
            (   "UPN"       "LEN"   )
            (   "FLT"
                (   "log_or"
                    (
                        (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                        (   "log_eq"    (   "ROOM"  "WC"    )   )
                    )
                )
            )
            (   "FLT_IN"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "IUS_T"     (   "log_eq"    (   "ID"    "T1"    )   )   )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
```

```
        (   "PLC_I1"
            (
                (   "log_not"
                    (   "log_eq"    (   "BASE"      1   )   )
                )
                (   "log_eq"    (   "BASE"      1   )   )
                (   1.0 )
                (   1.0 )
                100.0
            )
        )
        (   "IUS_STR"   (   "log_eq"    (   "ID"    "STAIR" )   )   )
        (   "PLC_STR"
            (
                (   "log_not"
                    (   "log_ism"   (   "BUILT"     1   )   )
                )
                (   "log_ism"   (   "ATTR"      "COR"   )   )
                (   1.0  2.0  3.0  4.0 )
                (   1.0  -1.0   )
                130.0
            )
        )
        (   "CPL_I1"    (   (   "IID"   "I1"    )   )   )
        (   "ID_1"      "I1"    )
        (   "ID_2"      "I2"    )
        (   "ID_3"      "I3"    )
        (   "ID_4"      "I4"    )
        (   "ID_5"      "I5"    )
        (   "ID_6"      "I6"    )
        (   "ID_7"      "I7"    )
        (   "ID_8"      "I8"    )
        (   "OUT_VAL"   0.0     )
        (   "A_CL"  (   "log_ism"   (   "BUILT"     1   )   )   )
        (   "C_MIL"     50000000.0          )
        (   "C2"        2.0                 )
        (   "CI0"       0                   )
        (   "CI1"       1                   )
        (   "M_INI"     (   2   1   2   )   )
        (   "CI"        "i"                 )
        (   "CL"        "l"                 )
        (   "L0"        ()                  )
    )
    (                                           ;   TEST_P REFERENCE SECTION
        (   3.0
            ()
            (   "PLC_I1"    "IUS_T"     "FLT_IN"    "CPL_I1"    )
            (   "P_IT1"     "IT1_CS"    )
        )
        (   4.0
            "G_IUPR"
            (   "FSTR"      "P_IT1"     "UPN"   )
            (   "C_LEN_1"       )
        )
        (   4.0
            "G_MLT"
            (   "FSTR"      "C_LEN_1"   "C2"    )
            (   "C_LEN_2"   )
        )
        (   2.0
            "PLACE"
            (   "IT1_CS"    "ID_1"      "ID_2"  "C_LEN_2"   "M_INI" )
            (   "CS_2"      "M_ID2" )
        )
        (   2.0
            "PLACE"
            (   "CS_2"      "ID_2"      "ID_3"  "C_LEN_2"   "M_ID2" )
            (   "CS_3"      "M_ID3" )
        )
```

```
                (   2.0
                    "PLACE"
                    (   "CS_3"      "ID_3"      "ID_4"  "C_LEN_2"   "M_ID3" )
                    (   "CS_4"      "M_ID4" )
                )
                (   2.0
                    "PLACE"
                    (   "CS_4"      "ID_4"      "ID_5"  "C_LEN_2"   "M_ID4" )
                    (   "CS_5"      "M_ID5" )
                )
                (   2.0
                    "PLACE"
                    (   "CS_5"      "ID_5"      "ID_6"  "C_LEN_2"   "M_ID5" )
                    (   "CS_6"      "M_ID6" )
                )
                (   2.0
                    "PLACE"
                    (   "CS_6"      "ID_6"      "ID_7"  "C_LEN_2"   "M_ID6" )
                    (   "CS_7"      "M_ID7" )
                )
                (   2.0
                    "PLACE"
                    (   "CS_7"      "ID_7"      "ID_8"  "C_LEN_2"   "M_ID7" )
                    (   "CS_8"      "M_ID8" )
                )
                (   3.0
                    ()
                    (   "PLC_STR"   "IUS_STR"   "FLT_IN"    "L0"    )
                    (   "P_STR"     "STR_CS"    )
                )
                (   4.0
                    "G_AREA"
                    (   "FSTR"  "A_CL"      "CI0"   "CI1"   "CI"    "CL"    )
                    (   "AREA"  )
                )
                (   4.0
                    "G_DVS"
                    (   "FSTR"  "C_MIL"     "AREA"  )
                    (   "VAL"   )
                )
            )
            (   "VAL"   )                               ;   TEST_P OUTPUT SECTION
        )
    )
```

The Unit **PLACE** performs the positioning of an Instance, name it ***NI***, of the Unit **TYPE**. ***NI*** should be adjacent to a previously positioned Instance, name it ***PI***, of the Unit **TYPE**. There are four positioning methods, each one having an integer as an identity. One or two of these methods are available for the positioning of ***NI***, depending on the history of the **TYPE** Instances' positioning.

The property **M_PL**, defined in the property section, is assigned a list of pairs. The first element of each pair, which constitutes the property name, is the triplet of the positioning method identities of the three last positioned **TYPE** Instances. This triplet will be referred to as the "positioning history". The second element, which constitutes the property value, is the list of the positioning methods, viewed as Unit Definition names, compatible with the positioning history. All the potential histories are represented in **M_PL**. Note that the Unit Definition name of each positioning method ends with the method's identity. The construction process, formally defined in the reference section, is presented here:

Input Terms

OCS_P	is the OCS of ***PI***
ID_P	is a string representing an identity for ***PI***
ID_N	is a string representing an identity for ***NI***

C_LEN_2 is the double length of the apartment type represented by the Unit **TYPE**
M_PREV is the current "positioning history"

Output Terms

OCS_N is the OCS of ***NI***
M_USED is the "positioning history" after the positioning of ***NI***

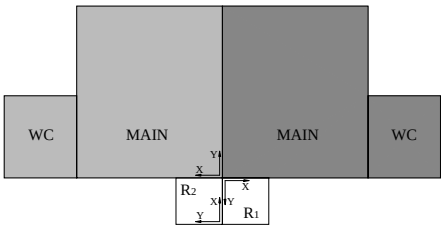
- Call the secondary formula **G_PL** to select the Unit Definition names of the positioning methods compatible to **M_PREV**. Name "**M_PAIR**" the list of these names.
- Call the secondary formula **G_MEM** to select one of these names at random. Name "**M_CURR**" the selected name.
- Call the secondary formula **G_SBS** to substitute the element "**O_NAME**" (which constitutes the value of **O_NAME_N**) with **M_CURR** in the list **CALL_L_S**, where both **O_NAME_N** and **CALL_L_S** are defined in the property section. Name "**CALL_L**" the resulting list. **CALL_L** constitutes a reference to the Unit Definition whose name is the value of **M_CURR**. The latter Unit Definition represents a positioning method.
- Call the high order formula (see § 2.2.e) to execute **CALL_L**. The execution results to the positioning of a new **TYPE** Instance (see the **METHOD** Unit Definitions, later in the code). Name "**OCS_N**" the new Instance's OCS and "**M_ID_N**" the identity of the positioning method used.
- Call the secondary formula **G_BF** to remove the first element of the list **M_PREV** and the formula **G_ADD** to append **M_ID_N** as the last element of the list. Name "**M_USED**" the resulting list which represents the new "positioning history".

```
(   "PLACE"                         ; --------------   UNIT PLACE
    (   "G_AND"
        (   "OCS_P"  "ID_P"  "ID_N"  "C_LEN_2"
            "M_PREV"    )                               ;   PLACE INPUT SECTION
        (                                               ;   PLACE PROP. SECTION
            (   "FLT"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "FSTR"      "at"    )
            (   "M_PL"
                (
                    ((  1   2   1   )   (   "METHOD2"   "METHOD3"   ))
                    ((  3   2   1   )   (   "METHOD2"   "METHOD3"   ))
                    ((  2   4   1   )   (   "METHOD2"   "METHOD3"   ))
                    ((  4   4   1   )   (   "METHOD2"               ))
                    ((  2   1   2   )   (   "METHOD1"   "METHOD4"   ))
                    ((  4   1   2   )   (   "METHOD1"   "METHOD4"   ))
                    ((  1   3   2   )   (   "METHOD1"   "METHOD4"   ))
                    ((  3   3   2   )   (   "METHOD1"   "METHOD4"   ))
                    ((  2   1   3   )   (   "METHOD2"   "METHOD3"   ))
                    ((  4   1   3   )   (   "METHOD2"               ))
                    ((  1   3   3   )   (   "METHOD2"               ))
                    ((  1   2   4   )   (   "METHOD1"   "METHOD4"   ))
                    ((  3   2   4   )   (   "METHOD1"   "METHOD4"   ))
                    ((  2   4   4   )   (   "METHOD1"               ))
                )
            )
            (   "O_NAME_N"      "O_NAME"    )
```

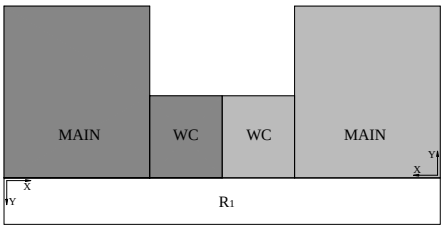
```
            (   "CALL_L_S"
                (   2.0
                    "O_NAME"
                    (   "OCS_P"  "ID_P"  "ID_N"  "C_LEN_2"  )
                    (   "OCS_N"  "M_USED"   )
                )
            )
        )
        (                                               ;   PLACE REFERENCE SECTION
            (   4.0
                "G_PL"
                (   "FSTR"      "M_PL"      "M_PREV"    )
                (   "M_PAIR"    )
            )
            (   4.0
                "G_MEM"
                (   "FSTR"      "M_PAIR"    )
                (   "M_CURR"    )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "CALL_L_S"      "O_NAME_N"  "M_CURR"    )
                (   "CALL_L"    )
            )
            (
                5.0
                ()
                (   "CALL_L"    )
                (   "OCS_N"     "M_ID_N"    )
            )
            (
                4.0
                "G_BF"
                (   "FSTR"      "M_PREV"    )
                (   "MPL"       )
            )
            (
                4.0
                "G_ADD"
                (   "FSTR"      "MPL"    "M_ID_N"   )
                (   "M_USED"    )
            )
        )
        (   "OCS_N"  "M_USED"   )                   ;   PLACE OUTPUT SECTION
    )
)
```

Each **METHOD** Unit (Figure 5) describes a deterministic method for positioning a **TYPE** Instance, name it ***NI***, given the positioning of an already placed **TYPE** Instance, name it ***PI***.

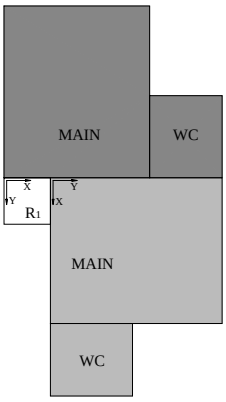
The positioning process includes the positioning of one or two fixed dimension Primitives referred to as the "connective Primitives" and the positioning of the Instance ***NI***. The "connective Primitives" have a property named **ATTR** assigned the value "**COR**" which is a spatial attribute that denotes a corridor and characterizes the region of the output map occupied by the Primitives. This is used as the base-region for the positioning of ***NI***. The construction process, formally defined in the reference section, is presented here:



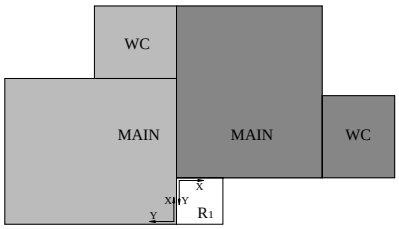
METHOD 1



METHOD 2



METHOD 3



METHOD 4

Figure 5

*Graphic representation of the positioning methods with respect to the homonymous Units. The Instance **PI** is represented as a dark shaded region, the Instance **NI** is represented as a light shaded region and the "connective Primitives" are represented as white regions. The OCS of each "connective Primitive" and the OCS of each **NI** Instance are presented as pairs of vectors indicating the OCS's axes.*

Input Terms

OCS_P	is the OCS of ***PI***
ID_P	is a string representing the identity of ***PI***
ID_N	is a string representing the identity of ***NI***
C_LEN	is the double length of the apartment type represented by the Unit **TYPE**

Output Terms

IT1_CS	is the OCS of ***NI***
M_ID	is the identity of the positioning method for ***NI***

Unit properties

PL_C_S	is the property list of a "connective Primitive"
PLC_C_R_S	is a "connective Primitive's" Positioning List described in the OCS of ***PI***. The names "**PLC_C1_R_S**" and "**PLC_C2_R_S**" substitute the name "**PLC_C_R_S**" in the Unit **METHOD1** which includes two "connective Primitives".
I_ID_S	is the Instance List (see § 2.2.c) of ***NI***
PLC_I_R_S	is the Positioning List of ***NI*** with respect to the OCS of ***PI***

- Call the secondary formula **G_SBS** to substitute any occurrence of the string "**ID_P**", which constitutes the value of the Unit's property **ID_P_N**, with the value **ID_P** in the lists **PL_C_S**, **PLC_C_R_S** and **PLC_I_R_S**.
- Call the secondary formula **G_SBS** to substitute any occurrence of the string "**ID_N**", which constitutes the value of the Unit's property **ID_N_N**, with the value **ID_N** in the list **I_ID_S**.
- Call the secondary formula **G_AMUCS** to transform the Positioning Lists from **OCS_P** to the World coordinate system (see § 1.6).
- Create the "connective Primitives".
- Create ***NI***

```
(   "METHOD1"                       ; --------------   UNIT METHOD1
    (   "G_AND"
        (   "OCS_P"   "ID_P"    "ID_N"  "C_LEN" )   ;   METHOD1 INPUT SECTION
        (                                           ;   METHOD1 PROP. SECTION
            (   "M_ID"      1       )
            (   "ID_P_N"    "ID_P"  )
            (   "ID_N_N"    "ID_N"  )
            (   "FLT"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "PL_C_S"
                (
                    (   "BUILT"     1       )
                    (   "ATTR"      "COR"   )
                    (   "ATTR"      "ID_P"  )
                )
            )
            (   "PLC_C1_R_S"
                (
                    (   "log_not"
                        (   "log_or"
                            (
                                (   "log_ism"   (   "IID"   "ID_P"  )   )
                                (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                                (   "log_eq"    (   "ROOM"  "WC"    )   )
                            )
                        )
                    )

                    (   "log_ism"   (   "IID"   "ID_P"  )   )
                    (   3.0     )
                    (   -1.0    )
                    130.0
                )
            )
```

```
            (   "PLC_C2_R_S"
                (
                    (   "log_not"
                        (   "log_or"
                            (
                                (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                                (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                                (   "log_eq"    (   "ROOM"  "WC"    )   )
                            )
                        )
                    )
                    (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                    (   2.0     )
                    (   1.0     )
                    130.0
                )
            )
            (   "PLC_I_R_S"
                (
                    (   "log_and"
                        (
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                            )
                            (   "log_not"
                                (   "log_ism"   (   "BUILT" 1       )   )
                            )
                        )
                    )
                    (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                    (    1.0    )
                    (   -1.0    )
                    130.0
                )
            )
            (   "FSTR"      "at"    )
            (   "L_COS"     (  1.0  0.0  -1.0  0.0  )  )
            (   "C1"        1.0     )
            (   "C4"        4.0     )
            (   "DIM_C"
                (
                    (   "plc_ddx_std"   )
                    (   130.0   130.0   )
                    (   "plc_ddx_std"   )
                    (   130.0   130.0   )
                )
            )
            (   "IUS_T"     (   "log_eq"    (   "ID"    "T1"    )   )   )
            (   "I_ID_S"        (   (   "IID"   "ID_N"  )   )   )
        )
        (                                       ;   METHOD1 REFERENCE SECTION
            (   4.0
                "G_SBS"
                (   "FSTR"      "PL_C_S"        "ID_P_N"    "ID_P"  )
                (   "PL_C"  )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "PLC_C1_R_S"    "ID_P_N"    "ID_P"  )
                (   "PLC_C1_R"  )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "PLC_C2_R_S"    "ID_P_N"    "ID_P"  )
                (   "PLC_C2_R"  )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "PLC_I_R_S"     "ID_P_N"    "ID_P"  )
                (   "PLC_I_R"   )
            )
```

```
            (   4.0
                "G_SBS"
                (   "FSTR"      "I_ID_S"        "ID_N_N"    "ID_N"  )
                (   "I_ID"  )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"      "PLC_C1_R"  "OCS_P"     "L_COS"   "C1"   "C4")
                (   "PLC_C1"    )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"      "PLC_C2_R"  "OCS_P"     "L_COS"   "C1"   "C4")
                (   "PLC_C2"    )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"      "PLC_I_R"   "OCS_P"     "L_COS"   "C1"   "C4")
                (   "PLC_I"     )
            )
            (   1.0
                ()
                (   "PL_C"      "PLC_C1"    "DIM_C"     )
                (   "MC1X"      "MC1Y"      "MC1_CS"    )
            )
            (   1.0
                ()
                (   "PL_C"      "PLC_C2"    "DIM_C"     )
                (   "MC2X"      "MC2Y"      "MC2_CS"    )
            )
            (   3.0
                ()
                (   "PLC_I"     "IUS_T"     "FLT"   "I_ID"      )
                (   "P_IT1"     "IT1_CS"    )
            )
        )
        (   "IT1_CS"    "M_ID"      )               ;   METHOD1 OUTPUT SECTION
    )
)


(   "METHOD2"                       ; --------------    UNIT METHOD2
    (   "G_AND"
        (   "OCS_P"   "ID_P"    "ID_N"  "C_LEN" )   ;   METHOD2 INPUT SECTION
        (                                           ;   METHOD2 PROP. SECTION
            (   "M_ID"      2       )
            (   "ID_P_N"    "ID_P"  )
            (   "ID_N_N"    "ID_N"  )
            (   "C_LEN_N"   "C_LEN" )
            (   "FLT"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "PL_C_S"
                (
                    (   "BUILT"     1       )
                    (   "ATTR"      "COR"   )
                    (   "ATTR"      "ID_P"  )
                )
            )
```

```
        (   "PLC_C_R_S"
            (
                (   "log_not"
                    (   "log_or"
                        (
                            (   "log_ism"   (   "IID"   "ID_P"  )   )
                            (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                            (   "log_eq"    (   "ROOM"  "WC"    )   )
                        )
                    )
                )
                (   "log_ism"   (   "IID"   "ID_P"  )   )
                (   3.0     )
                (   -1.0    )
                130.0
            )
        )
        (   "PLC_I_R_S"
            (
                (   "log_and"
                    (
                        (   "log_not"
                            (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                        )
                        (   "log_not"
                            (   "log_ism"   (   "BUILT" 1       )   )
                        )
                    )
                )
                (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                (    1.0    )
                (   -1.0    )
                130.0
            )
        )
        (   "FSTR"      "at"    )
        (   "L_COS"     (  1.0  0.0  -1.0  0.0  )  )
        (   "C1"        1.0     )
        (   "C4"        4.0     )
        (   "DIM_C_S"
            (
                (   "plc_ddx_std"       )
                (   "C_LEN"     "C_LEN" )
                (   "plc_ddx_std"       )
                (   130.0       130.0   )
            )
        )
        (   "IUS_T"     (   "log_eq"    (   "ID"    "T1"    )   )   )
        (   "I_ID_S"        (   (   "IID"   "ID_N"  )   )   )
    )
    (                                       ;   METHOD2 REFERENCE SECTION
        (   4.0
            "G_SBS"
            (   "FSTR"      "PL_C_S"        "ID_P_N"    "ID_P"  )
            (   "PL_C"  )
        )
        (   4.0
            "G_SBS"
            (   "FSTR"      "PLC_C_R_S"     "ID_P_N"    "ID_P"  )
            (   "PLC_C_R"   )
        )
        (   4.0
            "G_SBS"
            (   "FSTR"      "DIM_C_S"       "C_LEN_N"   "C_LEN" )
            (   "DIM_C" )
        )
```

```
            (   4.0
                "G_SBS"
                (   "FSTR"      "PLC_I_R_S"     "ID_P_N"    "ID_P"  )
                (   "PLC_I_R"   )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "I_ID_S"        "ID_N_N"    "ID_N"  )
                (   "I_ID"  )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"      "PLC_C_R"   "OCS_P"     "L_COS"  "C1"  "C4")
                (   "PLC_C"     )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"      "PLC_I_R"   "OCS_P"     "L_COS"  "C1"  "C4")
                (   "PLC_I"     )
            )
            (   1.0
                ()
                (   "PL_C"      "PLC_C"     "DIM_C"     )
                (   "MCX"       "MCY"       "MC_CS"     )
            )
            (   3.0
                ()
                (   "PLC_I"     "IUS_T"     "FLT"   "I_ID"      )
                (   "P_IT1"     "IT1_CS"    )
            )
        )
        (   "IT1_CS"    "M_ID"      )               ;   METHOD2 OUTPUT SECTION
    )
)



(   "METHOD3"                       ; --------------   UNIT METHOD3
    (   "G_AND"
        (   "OCS_P"   "ID_P"    "ID_N"  "C_LEN" )   ;   METHOD3 INPUT SECTION
        (                                           ;   METHOD3 PROP. SECTION
            (   "M_ID"      3       )
            (   "ID_P_N"    "ID_P"  )
            (   "ID_N_N"    "ID_N"  )
            (   "FLT"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "PL_C_S"
                (
                    (   "BUILT"     1       )
                    (   "ATTR"      "COR"   )
                    (   "ATTR"      "ID_P"  )
                )
            )
            (   "PLC_C_R_S"
                (
                    (   "log_not"
                        (   "log_or"
                            (
                                (   "log_ism"   (   "IID"   "ID_P"  )   )
                                (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                                (   "log_eq"    (   "ROOM"  "WC"    )   )
                            )
                        )
                    )
                    (   "log_ism"   (   "IID"   "ID_P"  )   )
                    (   3.0     )
                    (   -1.0    )
```

```
                130.0
            )
        )
        (   "PLC_I_R_S"
            (
                (   "log_and"
                    (
                        (   "log_not"
                            (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                        )
                        (   "log_not"
                            (   "log_ism"   (   "BUILT" 1       )   )
                        )
                    )
                )
                (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                (   4.0     )
                (   1.0     )
                130.0
            )
        )
        (   "FSTR"      "at"    )
        (   "L_COS"     (  1.0  0.0  -1.0  0.0  )  )
        (   "C1"        1.0     )
        (   "C4"        4.0     )
        (   "DIM_C"
            (
                (   "plc_ddx_std"   )
                (   130.0   130.0   )
                (   "plc_ddx_std"   )
                (   130.0   130.0   )
            )
        )
        (   "IUS_T"     (   "log_eq"    (   "ID"    "T1"    )   )   )
        (   "I_ID_S"        (   (   "IID"   "ID_N"  )   )   )
    )
    (                                       ;   METHOD3 REFERENCE SECTION
        (   4.0
            "G_SBS"
            (   "FSTR"      "PL_C_S"        "ID_P_N"    "ID_P"  )
            (   "PL_C"  )
        )
        (   4.0
            "G_SBS"
            (   "FSTR"      "PLC_C_R_S"     "ID_P_N"    "ID_P"  )
            (   "PLC_C_R"   )
        )
        (   4.0
            "G_SBS"
            (   "FSTR"      "PLC_I_R_S"     "ID_P_N"    "ID_P"  )
            (   "PLC_I_R"   )
        )
        (   4.0
            "G_SBS"
            (   "FSTR"      "I_ID_S"        "ID_N_N"    "ID_N"  )
            (   "I_ID"  )
        )
        (   4.0
            "G_AMUCS"
            (   "FSTR"      "PLC_C_R"   "OCS_P"     "L_COS"  "C1"  "C4")
            (   "PLC_C"     )
        )
        (   4.0
            "G_AMUCS"
            (   "FSTR"      "PLC_I_R"   "OCS_P"     "L_COS"  "C1"  "C4")
            (   "PLC_I"     )
        )
```

```
            (   1.0
                ()
                (   "PL_C"      "PLC_C"     "DIM_C"     )
                (   "MCX"       "MCY"       "MC_CS"     )
            )
            (   3.0
                ()
                (   "PLC_I"     "IUS_T"     "FLT"   "I_ID"      )
                (   "P_IT1"     "IT1_CS"    )
            )
        )
        (   "IT1_CS"    "M_ID"      )               ;   METHOD3 OUTPUT SECTION
    )
)


(   "METHOD4"                       ; --------------   UNIT METHOD4
    (   "G_AND"
        (   "OCS_P"   "ID_P"    "ID_N"  "C_LEN" )   ;   METHOD4 INPUT SECTION
        (                                           ;   METHOD4 PROP. SECTION
            (   "M_ID"      4       )
            (   "ID_P_N"    "ID_P"  )
            (   "ID_N_N"    "ID_N"  )
            (   "FLT"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "PL_C_S"
                (
                    (   "BUILT"     1       )
                    (   "ATTR"      "COR"   )
                    (   "ATTR"      "ID_P"  )
                )
            )
            (   "PLC_C_R_S"
                (
                    (   "log_not"
                        (   "log_or"
                            (
                                (   "log_ism"   (   "IID"   "ID_P"  )   )
                                (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                                (   "log_eq"    (   "ROOM"  "WC"    )   )
                            )
                        )
                    )
                    (   "log_ism"   (   "IID"   "ID_P"  )   )
                    (   3.0     )
                    (   -1.0    )
                    130.0
                )
            )
            (   "PLC_I_R_S"
                (
                    (   "log_and"
                        (
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                            )
                            (   "log_not"
                                (   "log_ism"   (   "BUILT" 1       )   )
                            )
                        )
                    )
                    (   "log_ism"   (   "ATTR"  "ID_P"  )   )
                    (   2.0     )
                    (   1.0     )
                    130.0
                )
            )
```

```
            (   "FSTR"      "at"    )
            (   "L_COS"     (  1.0  0.0  -1.0   0.0  )  )
            (   "C1"        1.0     )
            (   "C4"        4.0     )
            (   "DIM_C"
                (
                    (   "plc_ddx_std"  )
                    (   130.0   130.0   )
                    (   "plc_ddx_std"  )
                    (   130.0   130.0   )
                )
            )
            (   "IUS_T"     (   "log_eq"    (   "ID"    "T1"    )   )   )
            (   "I_ID_S"        (   (   "IID"   "ID_N"  )   )   )
        )
        (                                           ;   METHOD4 REFERENCE SECTION
            (   4.0
                "G_SBS"
                (   "FSTR"      "PL_C_S"        "ID_P_N"    "ID_P"  )
                (   "PL_C"  )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "PLC_C_R_S"     "ID_P_N"    "ID_P"  )
                (   "PLC_C_R"   )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "PLC_I_R_S"     "ID_P_N"    "ID_P"  )
                (   "PLC_I_R"   )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"      "I_ID_S"        "ID_N_N"    "ID_N"  )
                (   "I_ID"  )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"      "PLC_C_R"   "OCS_P"     "L_COS"  "C1"  "C4")
                (   "PLC_C"     )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"      "PLC_I_R"   "OCS_P"     "L_COS"  "C1"  "C4")
                (   "PLC_I"     )
            )
            (   1.0
                ()
                (   "PL_C"      "PLC_C"     "DIM_C"     )
                (   "MCX"       "MCY"       "MC_CS"     )
            )
            (   3.0
                ()
                (   "PLC_I"     "IUS_T"     "FLT"   "I_ID"      )
                (   "P_IT1"     "IT1_CS"    )
            )
        )
        (   "IT1_CS"    "M_ID"      )               ;   METHOD4 OUTPUT SECTION
    )
)
```

The Unit **TYPE_R** represents an apartment type having openings, furniture and bathroom fittings. The construction process, formally defined in the reference section, is presented here:

- Create a **TYPE** Instance adjacent to the central cell of the input map: the Unit **TYPE** (see the Unit Definition **TYPE**, later in the code) has a property named "**ID**", which stands for "identity", assigned the value "**T1**"; as a consequence it is identified by the Query **("log_eq" ("ID" "T1"))** represented by the property **IUS_T** defined in the property section of the Unit Definition **TYPE_R**. The Positioning List **PLC_I1**, also defined in the property section, defines a single placement of the Instance. The Instance List (see § 2.2.c), represented by the property **CPL_I1**, appends the value "**I1**" to the **IID** list (see Appendix B.2) of the output map's cells occupied by the Instance.
- Call the secondary formula **G_IUPR** to get the value of the property **LEN** of the instanced Unit, which specifies the length of the apartment type, i.e. the sum of the dimensions $\mathbf{R_X}$ and $\mathbf{W_X}$ appearing in Figure 4. Name "**C_LEN**" the length.
- Call the secondary formula **G_ADDPR** to insert a property named "**LEN**" (where "**LEN**" constitutes the value of the property **UPN**), having value **C_LEN**, in the **TYPE_R** Unit's property list.
- Call the Unit creation formulae **MAIN_FR** and **WC_FR** (see the respective Unit Definitions, later in the code) to place the openings, the furniture and the bathroom fittings of the main room and the WC respectively.

```
(   "TYPE_R"                        ; --------------   UNIT TYPE_R
    (   "G_AND"
        ()                                              ;   TYPE_R INPUT SECTION
        (                                               ;   TYPE_R PROP. SECTION
            (   "ID"        "TR"    )
            (   "FLT"
                (   "log_or"
                    (
                        (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                        (   "log_eq"    (   "ROOM"  "WC"    )   )
                    )
                )
            )
            (   "FLT_IN"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "IUS_T"     (   "log_eq"    (   "ID"    "T1"    )   )   )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "PLC_I1"
                (
                    (   "log_not"
                        (   "log_eq"    (   "BASE"      1   )   )
                    )
                    (   "log_eq"    (   "BASE"      1   )   )
                    (   1.0 )
                    (   1.0 )
                    100.0
                )
            )
            (   "FSTR"      "call"  )
            (   "UPN"       "LEN"   )
            (   "CL"        "l"     )
            (   "CPL_I1"    (   (   "IID"   "I1"    )   )   )
            (   "OUT_VAL"   0.0 )
        )
```

```
        (                                               ;   TYPE_R REFERENCE SECTION
            (   3.0
                ()
                (   "PLC_I1"    "IUS_T"     "FLT_IN"    "CPL_I1"    )
                (   "P_IT1"     "IT1_CS"    )
            )
            (   4.0
                "G_IUPR"
                (   "FSTR"      "P_IT1"     "UPN"   )
                (   "C_LEN"     )
            )
            (   4.0
                "G_ADDPR"
                (   "FSTR"      "UPN"   "C_LEN"     "CL"    )
                ()
            )
            (   2.0
                "MAIN_FR"
                ()
                (   "MFR_OUT"   )
            )
            (   2.0
                "WC_FR"
                ()
                (   "WCFR_OUT"  )
            )
        )
        (   "OUT_VAL"   )                           ;   TYPE_R OUTPUT SECTION
    )
)
```

The Unit **TYPE** represents an apartment type as an assembly of two Primitives. The first Primitive represents the main room while the second Primitive represents the WC of the apartment. The input terms of the Unit **TYPE** have been presented in § B.2. The construction process is presented here:

- Call the formula **G_HTYPE1** (see § B.3.1) to define the Sizing Lists of the Primitives.
- Create the first Primitive adjacent to the central cell of the Spatial Map. The Positioning List **PLC_MAIN**, appearing as an input term, defines a single positioning of the Primitive.
- Create the second Primitive adjacent to the first one. The Positioning List **PLC_WC** , appearing as an input term, defines a single positioning of the Primitive.
- Call the secondary formula **G_PLS** to calculate the length of the apartment type, name it "**CLENGTH**", as the sum of the dimensions **MX** and **WCX**, which respectively correspond to the dimensions $\mathbf{R_X}$ and $\mathbf{W_X}$ appearing in Figure 4.
- Call the secondary formula **G_ADDPR** to insert a property named "**LEN**" (where "**LEN**" constitutes the value of the property **UPN**), having value **CLENGTH**, in the **TYPE** Unit's property list.
- Return **CLENGTH**.

```
(   "TYPE"                      ; --------------    UNIT TYPE
    (   "G_AND"
        (                                           ;   TYPE INPUT SECTION
            "RX_MIN"    "RX_MAX"    "CWIDTH"
            "WX_MIN"    "WX_MAX"    "WE_MIN"    "WE_MAX"
            "PL_MAIN"   "PLC_MAIN"  "PL_WC"     "PLC_WC"
        )

        (                                           ;   TYPE PROP. SECTION
            (   "ID"        "T1"    )
            (   "FSTR"      "call"  )
            (   "L_STR"     (   "plc_ddx_std"   )   )
```

```
            (   "FLT"
                (   "log_or"
                    (
                        (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                        (   "log_eq"    (   "ROOM"  "WC"    )   )
                    )
                )
            )
            (   "ID_OCS"
                (   "log_eq"    (   "ROOM"      "MAIN"  )   )
            )
            (   "UPN"   "LEN"   )
            (   "CL"    "l"     )
        )
        (                                           ;   TYPE REFERENCE SECTION
            (   4.0
                "G_HTYPE1"
                (   "FSTR"      "L_STR"
                    "RX_MIN"    "RX_MAX"    "CWIDTH"
                    "WX_MIN"    "WX_MAX"    "WE_MIN"    "WE_MAX"
                )
                (   "DIM_MAIN"      "DIM_WC"    )
            )
            (   1.0
                ()
                (   "PL_MAIN"   "PLC_MAIN"  "DIM_MAIN"  )
                (   "MX"        "MY"        "M_CS"      )
            )
            (   1.0
                ()
                (   "PL_WC"     "PLC_WC"    "DIM_WC"    )
                (   "WCX"       "WCY"       "WC_CS"     )
            )
            (   4.0
                "G_PLS"
                (   "FSTR"  "MX"    "WCY"   )
                (   "CLENGTH"   )
            )
            (   4.0
                "G_ADDPR"
                (   "FSTR"  "UPN"   "CLENGTH"   "CL"    )
                ()
            )
        )
        (   "CLENGTH"   )                           ;   TYPE OUTPUT SECTION
    )
)
```

The Unit **MAIN_FR** includes the openings and the furniture of the main room of an apartment type. The construction process, formally defined in the reference section, is presented below. All the Positioning Lists, the Sizing Lists and the Queries appearing in the following presentation are defined in the property section of the Unit Definition **MAIN_FR**.

- Create an Instance of the standard Unit whose **ID** property has the value "**DOOR1**" (see § B.1 and Figure 3: standard Unit **DOOR**). The Unit's identification is performed by the Query **IUS_D1**. The **DOOR** Instance represents the entrance to the apartment. The Positioning List **PLC_DE** defines a single positioning of the Instance in the main room at the boundary with the non built space.
- Create a second Instance of the same standard Unit. This Instance represents the WC door. The Positioning List **PLC_DWC** defines two potential positioning possibilities for the Instance in the WC room at the boundary with the main room.

- Create an Instance of the standard Unit whose **ID** property has the value "**DOOR2**" (see § B.1 and Figure 3: standard Unit **DOOR-BALCONY**). The Unit's identification is performed by the Query **IUS_D2**. The Positioning List **PLC_DE** defines two potential positioning possibilities for the Instance in the main room at the boundary with the non built space.
- Create two Instances of the standard Unit whose **ID** property is assigned the value "**BED**" (see § B.1 and Figure 3: standard Unit **BED**). The Unit's identification is performed by the Query **IUS_BE**. The Positioning List **PLC_BED** allows each Instance to be placed in the main room, at the part of the room's boundary which is free from other objects (i.e. the **ATTR** list of the room space adjacent to the boundary is empty), at any orientation and mirroring condition.
- Create the Units **SIT** and **BOARD**. The first Unit represents a sitting place consisted of a small table and two chairs, while the second Unit represents a wardrobe. Both Units have input a Positioning List while the Unit **BOARD** has a Sizing List as the second input term. The Positioning Lists **PLC_SIT** and **PLC_BO**, which constitute input terms of the Units **SIT** and **BOARD** respectively, have the same logical interpretation with the Positioning List **PLC_BED**. The Sizing List **BO_DIM**, which constitutes the second input term of the Unit **BOARD**, allows the dimension of the wardrobe towards the positioning vector to vary in the range **[125.0, 200.0]** (see Appendix A.1: **plc_ddx_max**) in order to adapt to the available space, while the other dimension is defined to have the standard value **65.0** (see Appendix A.1: **plc_ddx_std**).

Note that the **FLT_IN** input term of each Instance construction formula (see § 2.2.c) is defined in the property section as an always confirmed Query, which means that all the Primitives of each Instance are imprinted to the output map of the Instance's creation formula. As a consequence the positioning of each Instance is constrained by the previously created Instances.

Note that the **FLT** property of the Unit **MAIN_FR** allows only the **FS** Primitive of the **DOOR** Instance (see § B.1 and Figure 3: standard Unit **DOOR**) to be imprinted on the Unit's output map.

```
(   "MAIN_FR"                       ; --------------   UNIT MAIN_FR
    (   "G_AND"
        ()                                          ;   MAIN_FR INPUT SECTION
        (                                           ;   MAIN_FR PROP. SECTION
            (   "ID"        "M_FR"  )
            (   "FLT"
                (   "log_eq"    (   "BLOCK"     "DOOR1" )   )
            )
            (   "FLT_IN"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "ID_OCS"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )

            (   "IUS_D1"    (   "log_eq"    (   "ID"    "DOOR1" )   )   )
            (   "IUS_D2"    (   "log_eq"    (   "ID"    "DOOR2" )   )   )
            (   "IUS_BE"    (   "log_eq"    (   "ID"    "BED"   )   )   )
            (   "PLC_DE"
                (
                    (   "log_eq"    (   "ROOM"      "MAIN"  )   )
                    (   "log_not"
                        ("log_ism"  (   "BUILT"     1   )   )
                    )
                    (   1.0 )
                    (   1.0 )
                    110.0
                )
            )
```

```
(   "PLC_DWC"
    (
        (   "log_eq"    (   "ROOM"      "WC"    )   )
        (   "log_eq"    (   "ROOM"      "MAIN"  )   )
        (   4.0 )
        (   1.0     -1.0    )
        110.0
    )
)
(   "PLC_DB"
    (
        (   "log_eq"    (   "ROOM"      "MAIN"  )   )
        (   "log_eq"    (   "BUILT"     ()      )   )
        (   3.0 )
        (   1.0     -1.0    )
        110.0
    )
)
(   "PLC_BED"
    (
        (   "log_and"
            (
                (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                (   "log_eq"    (   "ATTR"  ()      )   )
            )
        )
        (   "log_not"
            (   "log_eq"    (   "ROOM"  "MAIN"  )   )
        )
        (   1.0  2.0  3.0  4.0 )
        (   1.0 -1.0  )
        148.0
    )
)
(   "PLC_SIT"
    (
        (   "log_and"
            (
                (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                (   "log_eq"    (   "ATTR"  ()      )   )
            )
        )
        (   "log_not"
            (   "log_eq"    (   "ROOM"  "MAIN"  )   )
        )
        (   1.0  2.0  3.0  4.0 )
        (   1.0 -1.0  )
        126.0
    )
)
(   "PLC_BO"
    (
        (   "log_and"
            (
                (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                (   "log_eq"    (   "ATTR"  ()      )   )
            )
        )
        (   "log_not"
            (   "log_eq"    (   "ROOM"  "MAIN"  )   )
        )
        (   1.0  2.0  3.0  4.0 )
        (   1.0 -1.0   )
        120.0
    )
)
```

```
            (   "BO_DIM"
                (
                    (   "plc_ddx_max"   )
                    (   125.0   200.0   )
                    (   "plc_ddx_std"   )
                    (   65.0    65.0    )
                )
            )
            (   "L0"        ()  )
            (   "OUT_VAL"   0.0 )
        )
        (                                               ;   MAIN_FR REFERENCE SECTION
            (   3.0
                ()
                (   "PLC_DE"    "IUS_D1"    "FLT_IN"    "L0"    )
                (   "P_DE"      "DE_CS" )
            )
            (   3.0
                ()
                (   "PLC_DWC"   "IUS_D1"    "FLT_IN"    "L0"    )
                (   "P_DWC"     "DWC_CS"    )
            )
            (   3.0
                ()
                (   "PLC_DB"    "IUS_D2"    "FLT_IN"    "L0"    )
                (   "P_DB"      "DB_CS" )
            )
            (   3.0
                ()
                (   "PLC_BED"   "IUS_BE"    "FLT_IN"    "L0"    )
                (   "D_BED_1"   "BE_OCS_1"  )
            )
            (   3.0
                ()
                (   "PLC_BED"   "IUS_BE"    "FLT_IN"    "L0"    )
                (   "D_BED_2"   "BE_OCS_2"  )
            )
            (   2.0
                "SIT"
                (   "PLC_SIT"   )
                (   "S_2"   )
            )
            (   2.0
                "BOARD"
                (   "PLC_BO"    "BO_DIM"    )
                (   "BO_X"      )
            )
        )
        (   "OUT_VAL"   )                               ;   MAIN_FR OUTPUT SECTION
    )
)
```

The Unit **SIT** represents a sitting group that includes a table and two chairs. There are two standard Units representing such sitting places. The **ID** property of the first standard Unit is assigned the value "**SIT1**" while the **ID** property of the second standard Unit is assigned the value "**SIT3**" (see § B.1 and Figure 3: standard Units **SITTING GROUP 1** and **SITTING GROUP 2**). An Instance of one of these Units is created, due to the connective **G_OR**. The positioning of the Instance depends on the Positioning List **PLC_SIT** which constitutes the input term of the Unit **SIT**.

Note that the **FLT** property of the Unit **SIT** is always confirmed. As a consequence every Primitive of the Instanced Unit is imprinted to the output map of the Unit **SIT**.

```
(   "SIT"                          ; --------------   UNIT SIT
    (   "G_OR"
        (   "PLC_SIT"   )                              ;   SIT INPUT SECTION
        (                                              ;   SIT PROP. SECTION
            (   "ID"    "SIT"   )
            (   "IUS_S1"    (   "log_eq"    (   "ID"    "SIT1"  )   )   )
            (   "IUS_S2"    (   "log_eq"    (   "ID"    "SIT3"  )   )   )
            (   "FLT"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "ID_OCS"
                (   "log_not"
                    (   "log_eq"    (   "NONE"  1   )   )
                )
            )
        )
        (                                              ;   SIT REFERENCE SECTION
            (   3.0
                ()
                (   "PLC_SIT"   "IUS_S1"    "FLT"   "L0"    )
                (   "S_1"       "S_OCS_1"   )
            )
            (   3.0
                ()
                (   "PLC_SIT"   "IUS_S2"    "FLT"   "L0"    )
                (   "S_2"       "S_OCS_2"   )
            )
        )
        (   "S_2"   )                                  ;   SIT OUTPUT SECTION
    )
)
```

The Unit **BOARD** represents a wardrobe. It includes two Primitives (Figure 6): the first Primitive represents the physical object wardrobe while the second Primitive represents the free space associated with the physical object. The second Primitive lies at the top of the first one with respect to the first Primitive's OCS. Both Primitives have the same length since the top edge of the first Primitive coincides with the bottom edge of the second Primitive. The construction process, formally defined in the reference section, is presented here:

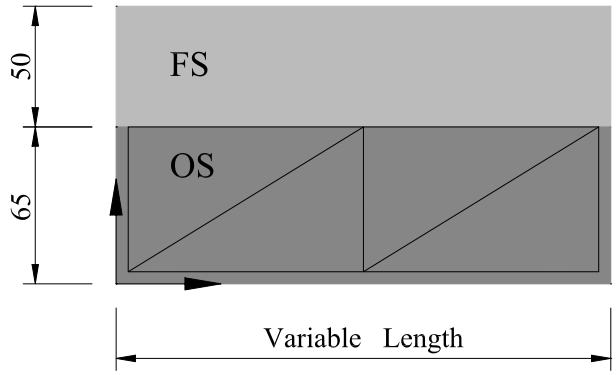


Figure 6

*The primitives of the Wardrobe Unit are represented as shaded rectangles. The dimensions of the Primitives represent centimeters. The spatial attributes **OS** and **FS**, each one assigned to a Primitive, denote "object space" and "free space" respectively. The OCS of the **OS** Primitive appears as a system of two vectors, perpendicular to each other, where the horizontal vector represents the X axis of the OCS. The drawing appearing on the **OS** Primitive is to be drawn by the CAD interface in the solution presentation phase.*

- Create the first Primitive with the Positioning List **PLC_OS** and the Sizing List **BO_DIM** which both constitute input terms of the Unit **BOARD**. This Primitive is assigned the property list **OS_PL**, defined in the property section, which includes: a) the property **PARAM** having the value **("BOARD")** which indicates that the wardrobe should be designed as a parametric object named "**BOARD**", b) the property **ATTR** having the value "**OS**" which represents the object's physical space and c) the property **APA** (see 2.2.a) having value an always confirmed Query.
- Call the secondary formula **G_AMUCS** to transform the Positioning List **PLC_FS_R** of the second Primitive from the OCS **OS_OCS** of the first Primitive to the World coordinate system (see § 1.6). The resulting Positioning List **PLC_FS** defines a single positioning of the second Primitive, in adjacency with the first one.
- Call the secondary formula **G_SBS** to substitute each occurrence of the string "**DX**" in the list **FS_DIM_S**, which is defined in the property section, with the length **OS_X** of the first Primitive. The resulting list **FS_DIM** is the Sizing List of the second Primitive.
- Create the second Primitive with the Positioning List **PLC_FS** and the Sizing List **FS_DIM**. This Primitive is assigned the property list **FS_PL**, defined in the property section, which includes the property **ATTR** assigned the value "**FS**" which represents the object's free space, and the property **APA** (see 2.2.a) having value an always confirmed Query.

```
(   "BOARD"                         ; --------------   UNIT BOARD
    (   "G_AND"
        (   "PLC_OS"    "BO_DIM"    )               ;   BOARD INPUT SECTION
        (                                           ;   BOARD PROP. SECTION
            (   "OS_PL"
                (
                    (   "PARAM"     (   "BOARD" )   )
                    (   "ATTR"      "OS"            )
                    (   "APA"
                        (   "log_not"
                            (   "log_eq"    (   "NONE"      1   )   )
                        )
                    )
                )
            )
            (   "FS_PL"
                (
                    (   "ATTR"      "FS"    )
                    (   "APA"
                        (   "log_not"
                            (   "log_eq"    (   "NONE"      1   )   )
                        )
                    )
                )
            )
            (   "PLC_FS_R"
                (
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"  "MAIN"  )   )
                            (   "log_not"
                                (   "log_ism"   (   "ATTR"  "OS"    )   )
                            )
                        )
                    )
                    (   "log_ism"   (   "PARAM"     "BOARD"     )   )
                    (   1.0  )
                    (   1.0  )
                    60.0
                )
            )
```

```
            (   "FLT"
                (   "log_not"
                    (   "log_eq"    (   "NONE"  1   )   )
                )
            )
            (   "ID_OCS"
                (   "log_eq"
                    (   "PARAM"     (   "BOARD" )   )
                )
            )
            (   "FSTR"      "at"    )
            (   "L_COS"     (  1.0  0.0  -1.0  0.0  )  )
            (   "C1"        1.0     )
            (   "C4"        4.0     )
            (   "DXN"       "DX"    )
            (   "FS_DIM_S"
                (
                    (   "plc_ddx_std"   )
                    (   "DX"    "DX"    )
                    (   "plc_ddx_std"   )
                    (   50.0    50.0    )
                )
            )
        )
        (                                           ;   BOARD REFERENCE SECTION
            (   1.0
                ()
                (   "OS_PL"     "PLC_OS"    "BO_DIM"    )
                (   "OS_X"      "OS_Y"      "OS_OCS"    )
            )
            (   4.0
                "G_AMUCS"
                (   "FSTR"  "PLC_FS_R"  "OS_OCS"    "L_COS"  "C1"  "C4")
                (   "PLC_FS"    )
            )
            (   4.0
                "G_SBS"
                (   "FSTR"  "FS_DIM_S"  "DXN"   "OS_X"  )
                (   "FS_DIM"    )
            )
            (   1.0
                ()
                (   "FS_PL"     "PLC_FS"    "FS_DIM"    )
                (   "FS_X"      "FS_Y"      "FS_OCS"    )
            )
        )
        (   "OS_X"  )                              ;   BOARD OUTPUT SECTION
    )
)
```

The Unit **WC_FR** includes the openings and the bathroom fittings of the WC of an apartment type. A WC window, a shower, a basin and a toilet, each one represented as an Instance of the respective standard Unit (see § B.1 and Figure 3), are successively created and positioned at the edges of the WC space. Each standard Unit is identified by its **ID** property, while the identification is performed by the Queries **IUS_WWC**, **IUS_DOU**, **IUS_BA** and **IUS_TO** respectively. The positioning specifications for the Instances are defined in the Positioning Lists **PLC_WWC**, **PLC_WC_DOU**, **PLC_WC_B** and **PLC_WC_TO** respectively. Both the aforementioned Queries and Positioning Lists are defined in the property section of the Unit Definition **WC_FR**.

Note that the **FLT_IN** input term of each Instance construction formula (see § 2.2.c) is defined in the property section as an always confirmed Query, which means that all the Primitives of each Instance are imprinted to the output map of the Instance's creation formula. As a consequence the positioning of each Instance is constrained by the previously created Instances.

First the window Instance is placed at the boundary of the WC space not occupied by the WC door with the non built space. Then the remaining three Instances are positioned at the boundary of the WC space. The basin Instance can be placed only at the part of the boundary which is free of any other object whereas the shower and the toilet Instances are not constrained by the window.

Note that the **FLT** property of the Unit **WC_FR** is assigned an always disconfirmed Query; as a consequence no imprinting is made on the Unit's output map.

```
(   "WC_FR"                         ; --------------   UNIT WC_FR
    (   "G_AND"
        ()                                              ;   WC_FR INPUT SECTION
        (                                               ;   WC_FR PROP. SECTION
            (   "ID"        "WC_FR" )
            (   "IUS_WWC"   (   "log_eq"    (   "ID"    "WWC"   )   )   )
            (   "IUS_DOU"   (   "log_eq"    (   "ID"    "DOUCHE")   )   )
            (   "IUS_BA"    (   "log_eq"    (   "ID"    "BASIN" )   )   )
            (   "IUS_TO"    (   "log_eq"    (   "ID"    "TOIL"  )   )   )
            (   "FLT_IN"
                (   "log_not"   (   "log_eq"    (   "NONE"  1   )   )   )
            )
            (   "FLT"
                (   "log_eq"    (   "NONE"      1   )   )
            )
            (   "ID_OCS"
                (   "log_not"
                    (   "log_eq"    (   "NONE"  1   )   )
                )
            )
            (   "PLC_WWC"
                (
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"      "WC"    )   )
                            (   "log_eq"    (   "ATTR"      ()      )   )
                        )
                    )
                    (   "log_eq"    (   "BUILT"     ()      )   )
                    (   3.0 )
                    (   1.0     -1.0    )
                    80.0
                )
            )
            (   "PLC_WC_TO"
                (
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"  "WC"    )   )
                            (   "log_or"
                                (
                                    (   "log_eq"    (   "ATTR"  ()  )   )
                                    (   "log_eq"
                                        (   "ATTR"  ( "WS" )    )
                                    )
                                )
                            )
                        )
                    )
                    (   "log_not"
                        (   "log_eq"    (   "ROOM"  "WC"    )   )
                    )
                    (   1.0  2.0  3.0  4.0 )
                    (   1.0  )
                    60.0
                )
            )
```

```
            (   "PLC_WC_DOU"
                (
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"  "WC"    )   )
                            (   "log_or"
                                (
                                    (   "log_eq"    (   "ATTR"  ()  )   )
                                    (   "log_eq"
                                        (   "ATTR"  ( "WS" )    )
                                    )
                                )
                            )
                        )
                    )
                    (   "log_not"
                        (   "log_eq"    (   "ROOM"  "WC"    )   )
                    )
                    (   2.0       )
                    (   1.0 -1.0  )
                    60.0
                )
            )
            (   "PLC_WC_B"
                (
                    (   "log_and"
                        (
                            (   "log_eq"    (   "ROOM"  "WC"    )   )
                            (   "log_eq"    (   "ATTR"  ()      )   )
                        )
                    )
                    (   "log_not"
                        (   "log_eq"    (   "ROOM"  "WC"    )   )
                    )
                    (   1.0  2.0  3.0  4.0 )
                    (   1.0 -1.0  )
                    70.0
                )
            )
            (   "L0"        ()  )
        )
        (                                           ;   WC_FR REFERENCE SECTION
            (   3.0
                ()
                (   "PLC_WWC"       "IUS_WWC"   "FLT_IN"    "L0"    )
                (   "P_WWC"         "WWC_CS"    )
            )
            (   3.0
                ()
                (   "PLC_WC_DOU"    "IUS_DOU"   "FLT_IN"    "L0"    )
                (   "E_DOU"         "E_DOU_OCS" )
            )
            (   3.0
                ()
                (   "PLC_WC_B"      "IUS_BA"    "FLT_IN"    "L0"    )
                (   "E_BA"          "E_BA_OCS"  )
            )
            (   3.0
                ()
                (   "PLC_WC_TO"     "IUS_TO"    "FLT_IN"    "L0"    )
                (   "E_TO"          "E_TO_OCS"  )
            )
        )
        (   "L0"    )                               ;   WC_FR OUTPUT SECTION
    )
)
```

B.4 The representation of the solution as an architectural drawing

The Unit **TEST_P** represents the solution of the design problem presented here. This Unit includes Instances of the Unit **TYPE** which represents an apartment type without furniture, openings and bathroom fittings. The full apartment type, having furniture, openings and bathroom fittings, is represented by the Unit **TYPE_R** which includes a **TYPE** Instance but it is not directly referenced in **TEST_P**. As a consequence the evolution of the Unit **TYPE** depends on both **TEST_P** and **TYPE_R** goals. This makes possible a compact problem representation, based on the interdependencies between the Units **TEST_P**, **TYPE** and **TYPE_R**, while computational time is saved since **TYPE_R** Instances (which are much more complex objects than the **TYPE** Instances) does not have to be included in **TEST_P**. However such an inclusion is necessary in the solution presentation phase, where the solution has to be represented as an architectural drawing showing fully equipped apartments.

A variation of the code presented in § B.3 was used for the solution presentation. Within this variation the value of the property **IUS_T** of each **METHOD** Unit was changed from **("log_eq" ("ID" "T1"))** to **("log_eq" ("ID" "TR"))** which specifies the Unit **TYPE_R**, instead of **TYPE**, to be instanced.

The CAD interface (see § 3.2) performing the solution presentation imprints all the terminal Primitives of the solution, ignoring the constraints posed by both the $\mathbf{P_{FLT}}$ Queries of the Instance creation formulae (see § 2.2.c) and the **FLT** Unit properties (see § 2.3).

The following actions have been programmed in the CAD interface:

- Draw thick walls at the boundaries between the regions specified by the Queries:

```
(  "log_or"
   (
       (   "log_eq"    (   "ROOM"  "MAIN"  )   )
       (   "log_eq"    (   "ROOM"  "WC"    )   )
       (   "log_ism"   (   "ATTR"  "COR"   )   )
   )
)
```

and

```
(  "log_not"
   (   "log_or"
       (
           (   "log_eq"    (   "ROOM"  "MAIN"  )   )
           (   "log_eq"    (   "ROOM"  "WC"    )   )
           (   "log_ism"   (   "ATTR"  "COR"   )   )
           (   "log_ism"   (   "ATTR"  "OPN"   )   )
       )
   )
)
```

- Draw thin walls at the boundaries between the regions specified by the Queries:

```
(  "log_and"
   (
       (   "log_eq"    (   "ROOM"  "MAIN"  )   )
       (   "log_not"
           (   "log_ism"   (   "ATTR"  "OPN"   )   )
       )
   )
)
```

and

```
(  "log_eq"    (   "ROOM"  "WC"    )   )
```

- Draw thin walls at the boundaries between the regions specified by the Queries:

```
(  "log_and"
   (
       (   "log_ism"   (   "ATTR"  "COR"   )   )
       (   "log_not"
           (   "log_ism"   (   "ATTR"  "OPN"   )   )
       )
   )
)
```

and

```
(  "log_or"
   (
       (   "log_eq"    (   "ROOM"  "MAIN"  )   )
       (   "log_eq"    (   "ROOM"  "WC"    )   )
   )
)
```

- Draw thin walls at the boundaries between regions having different values of the **IID** property

- Draw the fixed drawings assigned to the standard Units presented in Figure 3 and the parametric drawing assigned to the Wardrobe Unit presented in Figure 6.