

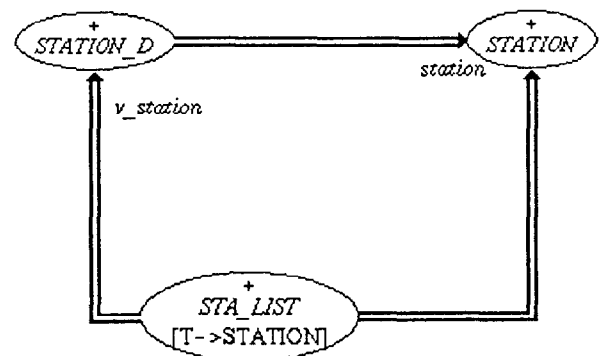
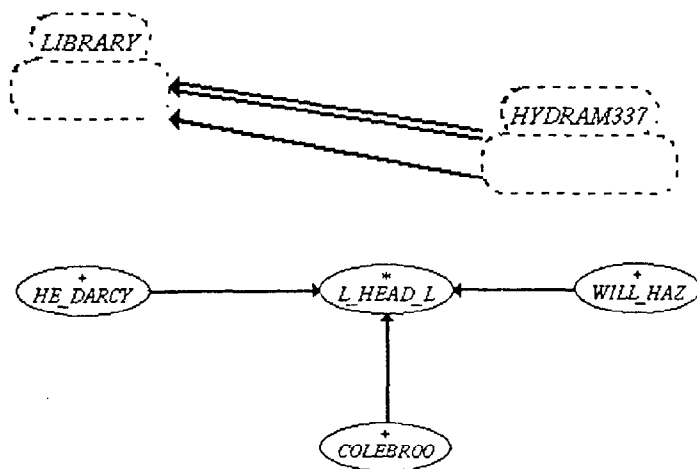


L'Institut
français
de recherche
scientifique
pour le
développement
en coopération

Burkina Faso

HYDRAM

Document technique sur HYDRAM Logiciel de Simulation des Hydro-aménagements.



Alain DEZETTER
Amidou OUEDRAOGO

Fonds Documentaire ORSTOM
Cote : *Ax M80* Ex : *1*

Ouagadougou, Août 1996

Fonds Documentaire ORSTOM
Cote : *Ax M80* Ex : *1*

1. Introduction

HYDRAM est un logiciel de simulation des Systèmes d'eau ou Hydro-aménagements ; il a été développé dans le langage Eiffel¹. Eiffel est un langage de programmation orienté objet ; un langage autonome avec des possibilités d'héritage multiple, et des mécanismes de développement rapides. Eiffel est à sa Version 3.3.7. Il est portable sur UNIX et WINDOWS 95. La version actuelle d'HYDRAM peut donc fonctionner sur UNIX ; et maintenant sur WINDOWS 95. Pour notre part, c'est la version sur UNIX qui fera l'objet de nos propos. L'univers d'HYDRAM compte 102 Clusters et 1095 Classes (y compris les Clusters et Classes des bibliothèques livrées par Eiffel). Il existe différents produits d'Eiffel, EiffelBench, EiffelCase, EiffelBuild... ; c'est avec les deux premiers que nous avons reconstitué le système HYDRAM ; à savoir EiffelBuild et principalement EiffelCase.

Dans ce document nous allons décrire le système d'HYDRAM :

- en ce qui concerne les classes d'HYDRAM ; nous passerons donc en revue leur organisation logique et physique, puis les types de liens (essentiellement des liens d'héritage, et des liens client)
- les différents Clusters du système qui décrivent en une première approche, de la conception, la notion de regroupement logique des Classes du système ; puis en une seconde approche beaucoup plus physique, la "localisation" physique des fichiers, incluant la notion de répertoire.
- l'ensemble des classes appelé ACE, qui décrit l'architecture du système ; c'est donc une "étiquette" identifiant le système d'HYDRAM
- enfin les annexes nous permettront d'appréhender le regroupement des Classes dans les différents Clusters du système d'HYDRAM.

Ce document a été essentiellement élaboré à partir de graphiques sortis d'EiffelCase ; un outil d'Eiffel, conçu pour la documentation des applications développées sur Eiffel.

Notre travail sera essentiellement axé sur les clusters et classes de l'application HYDRAM proprement dite ; c'est à dire, les classes qui ont été écrites et qui n'ont pas été fournies par Eiffel. Par le principe même de la programmation orientée objet, l'application HYDRAM dépend incontestablement des classes bibliothèques fournies par Eiffel dont nous ne ferons pas cas.

¹ Pour de plus amples informations consulter : "Eiffel, the language" de Bertrand MEYER édition Prentice-Hall, Object-Oriented Series.

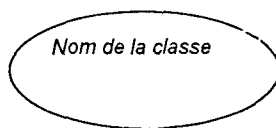
2. Définitions générales

2.1. Classes

Une classe est une implémentation d'un type abstrait ; c'est à dire qu'elle décrit un ensemble d'objets caractérisés par les opérations qui leurs sont applicables. Une classe est définie par ses méthodes (Procédures et Fonctions ou "*Routines*") et ses propriétés (Variables et Constantes ou "*Attributes*"). Un objet est une instance d'une classe. La notion de classe n'est perceptible qu'au moment du développement ou "*Compile-time*" et celle d'objet au moment de l'exécution d'une application ou "*Run-time*". Une classe ne peut donc être confondue à un objet.

En plus des propriétés d'héritage des classes (possibilités d'utiliser directement ou indirectement les "*Routines*" ou "*Attributes*" des classes dites Pères ou Ancêtres) ; une classe X peut être utilisée par d'autres classes ; on dira dans ce cas que ces classes sont des "*Clients*" de la classe X. Les types de liens que nous rencontrerons entre classes seront principalement des liens d'héritages et des liens clients. Ces deux notions (héritage multiple et clients) font d'Eiffel un langage de programmation orienté objet très efficace.

Dans la suite du document, nous allons utiliser la même convention de représentation schématique d'une classe sous la forme donnée par EiffelCase.



Nom de la classe représente le nom effectif que l'on a donné à la classe. Ce nom peut posséder un signe "+" dans ce cas on dira que la classe est effective ou d'un signe "*" dans ce cas on dira que la classe est différée

Ces possibilités offertes, imposent une organisation des classes dans des clusters.

2.2. Clusters

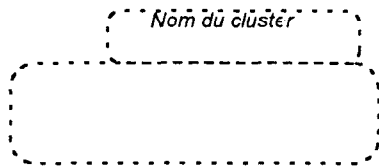
Nous allons définir un cluster suivant deux concepts :

- *Logique* : C'est un regroupement de plusieurs classes ou encore des clusters dans une structure arborescente. Ces classes et clusters devront avoir des traits communs ou de dépendance immédiats quand il s'agit des types de traitements qu'ils effectueront ; du moins à partir d'un niveau supérieur de l'arborescence. Plus on descend à la base de l'arborescence, plus les clusters se spécialisent donnant ainsi des regroupements de classes et de clusters spécifiques.

Cette organisation logique des classes et des clusters facilite la conception d'un système, et permet du coup une évolutivité aisée du système considéré.

- *Physique* : Après cette organisation logique, de subdivision des classes et clusters ; la réalisation du système passe par un stockage de fichiers sur support magnétique (disque dur). C'est alors que les clusters deviennent des répertoires, pouvant donc contenir d'autres répertoires et fichiers des classes.

Dans la suite du document, nous allons utiliser toujours la même convention de représentation schématique d'un cluster sous la forme donnée par EiffelCase.



Nom du cluster représente le nom effectif que l'on a donné au cluster.

2.3. Lien d'héritage

Un lien d'héritage entre deux classes, entre deux clusters, ou entre une classe et un cluster, est un symbole graphique, permettant de définir une propriété d'héritage (vue précédemment) entre classes, entre clusters ou entre classe et cluster.

Dans la suite du document, nous allons utiliser la même convention de représentation schématique d'un lien d'héritage sous la forme donnée par EiffelCase.

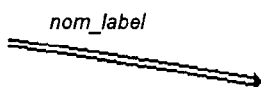


Un lien d'héritage sera symbolisé par une flèche orientée, dans le sens d'héritage fils à père.

2.4. Lien client

Un lien client entre deux classes, entre deux clusters, ou entre une classe et un cluster, est un symbole graphique, permettant de définir une propriété client (vue précédemment) entre classes, entre clusters ou entre classe et cluster.

Dans la suite du document, nous allons utiliser la même convention de représentation schématique d'un lien client sous la forme donnée par EiffelCase.



Un lien client sera symbolisé par une double flèche orientée, dans le sens client vers fournisseur.

Un lien client peut porter un label. Le label reste le plus souvent un nom de "attributes" ou de "routines".

2.5. Ace

Une application développée dans le langage Eiffel pour être exécutable, doit regrouper des classes. Toute application exécutable dans ce cas est appelé un **système** (du type programme dans certain langage comme le Pascal par exemple). L'ensemble des classes définissant le système est appelé **univers**. Une classe est désignée comme classe *racine* du système ; une procédure de cette classe est appelée procédure de *création* ou encore appelée procédure d'initialisation. Exécuter un système, c'est donc créer une instance directe de la classe racine (objet racine pour l'exécution courante), et lui appliquer la procédure de création ou d'initialisation. Cette procédure de création va créer les autres objets dont elle a besoin.

Les classes sont enregistrées dans des fichiers séparés avec le nom de la forme *nom.e*. Les regroupements des classes se font dans des clusters vus comme des répertoires.

Les classes d'un système sont composées de la classe racine et des autres classes auxquelles la classe racine fera appel directement ou indirectement. Spécifier un système revient donc à indiquer la liste des répertoires contenant les fichiers des classes, le nom de la classe racine (qui doit être une des classes de l'univers), et de la procédure de création (qui doit être une procédure de la classe racine de l'univers).

Cet ensemble de classes est appelé ACE (Assembly of Classes in Eiffel). Il est décrit dans un langage appelé LACE (Language for the Assembly of Classes in Eiffel). Il est matérialisé par un fichier portant le même nom c'est à dire ACE.

Le fichier ACE est ainsi indispensable pour la compilation.

Notion de variable d'environnement sur Eiffel: Une variable d'environnement est une variable ayant un contenu donné ; on peut donc accéder au contenu de la variable en faisant référence au nom de la variable.

Exemple : la variable d'environnement **HYD** = /home2/users/hydrum/hyd_0295.dir

On peut accéder au répertoire hyd_0295.dir en faisant **cd \$HYD**.

Le contenu d'un fichiers Ace a cette structure:

system

nom du système

root

nom de la classe racine (nom du cluster): "nom de la procédure de création"

default (permet de définir des paramètres par défaut qui seront pris en compte dans la compilation)

assertion (require);

precompiled ("EIFFEL3/precomp/spec/sparc/mvision");

cluster

nom du cluster 1 "chemin absolu du cluster 1";

nom du cluster 2 "chemin absolu du cluster 2";

..... "....."

..... "....."

```
..... "....."  
..... "....."  
end
```

Pour des raisons de commodités et de clarté dans la présentation de l'Ace, on regroupe les chemins d'accès des clusters par thème : les clusters d'EiffelVision sont regroupés différemment de ceux d'EiffelBase (par exemple).

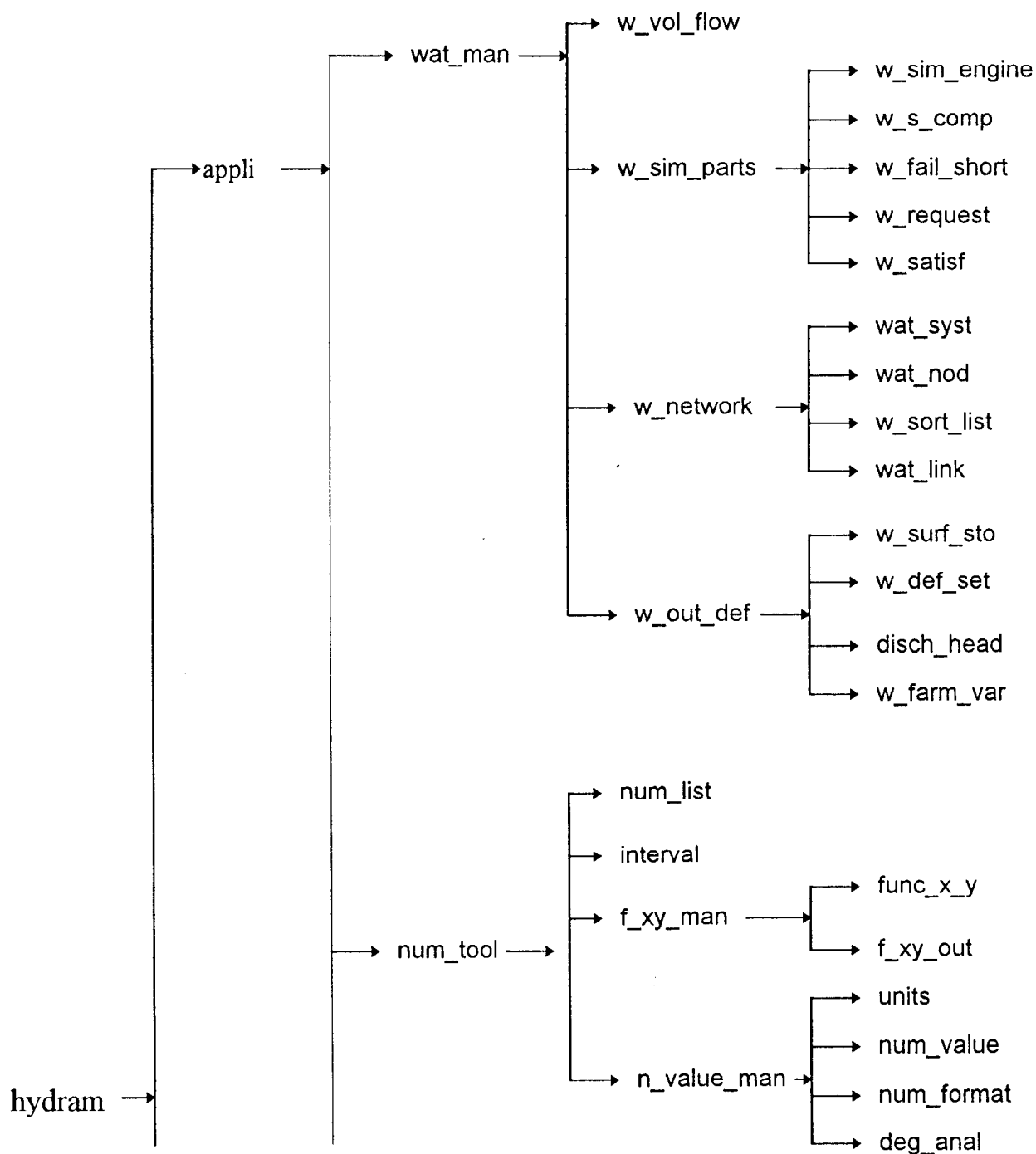
On a les regroupements suivant :

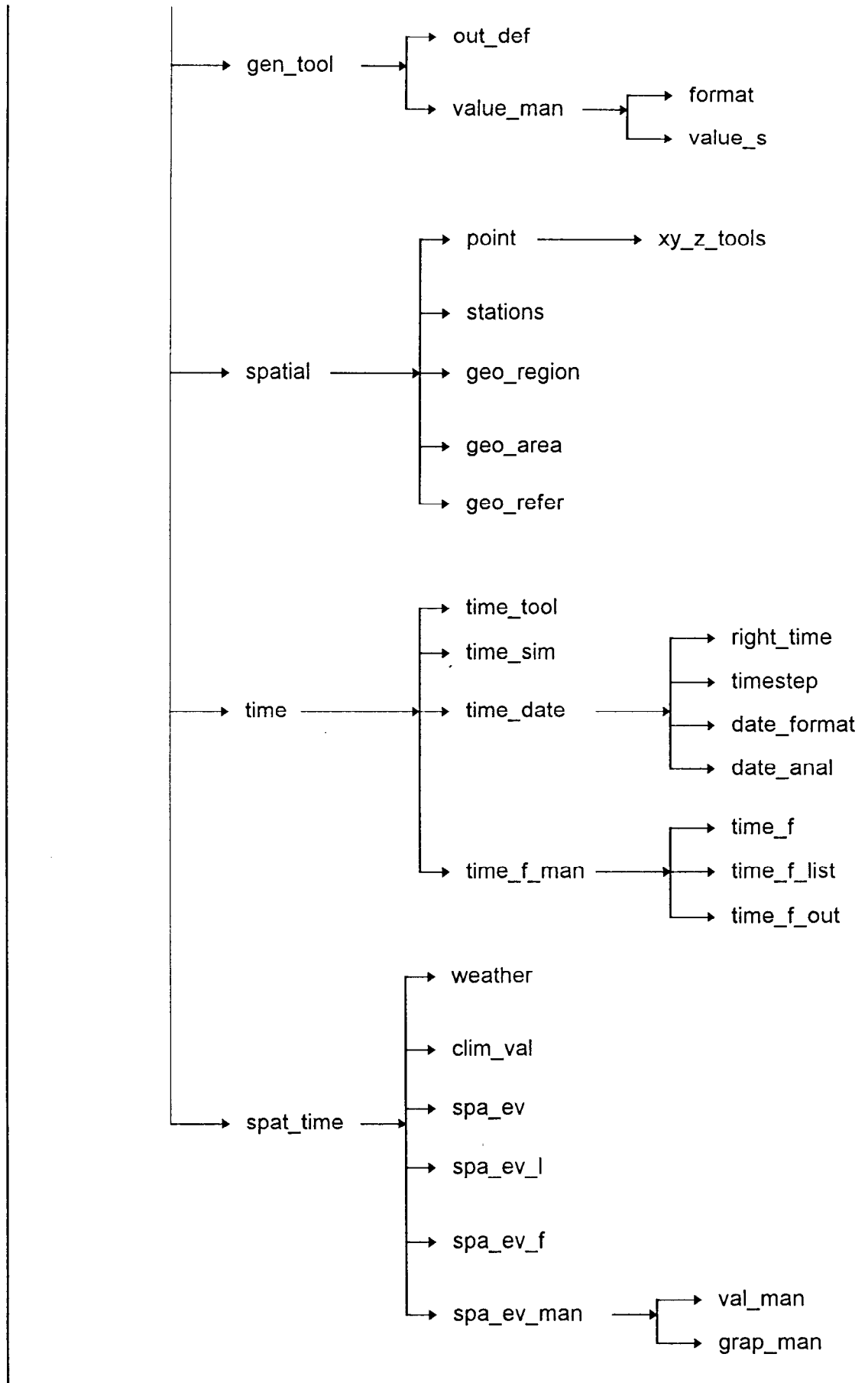
- Application ⇒ contient les classes concernant directement l'application HYDRAM (donc responsable de la définition des traitements).
- Interface ⇒ contient les classes directement liées à l'interface entre l'utilisateur et l'application.
- EiffelBase ⇒ contient les classes bibliothèques livrées par Eiffel ; formant la base pour le développement des classes d'une application.
- EiffelVision ⇒ contient les classes bibliothèques livrées par Eiffel ; formant la base pour le développement des classes de l'interface d'une application.
- External (objet extérieur) ⇒ permet d'intégrer des objets ne pouvant pas être décrits dans le langage Eiffel, ou déjà existants dans un autre langage compréhensible par Eiffel.

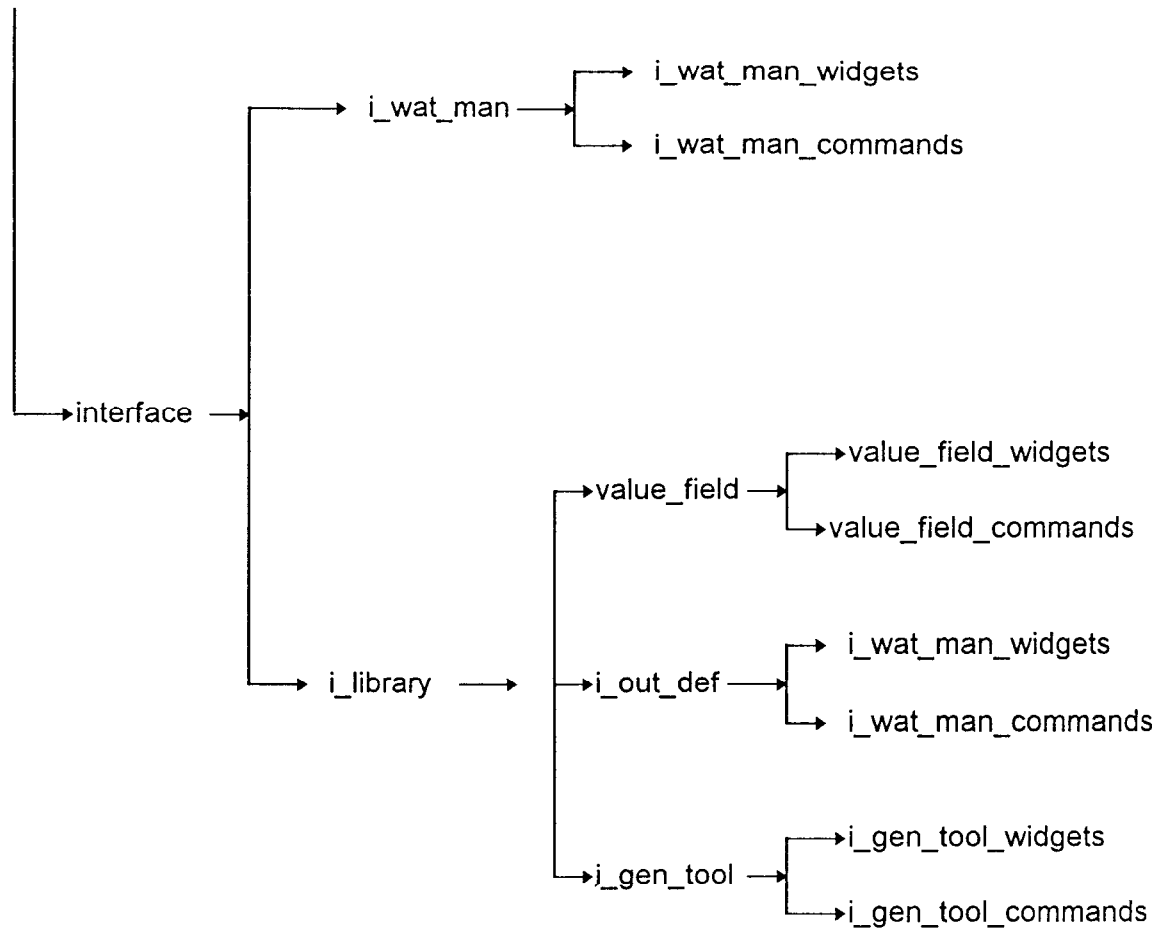
3. Liens d'héritage et client entre clusters

3.1. Arborescence des clusters d'HYDRAM

Les clusters d'HYDRAM sont essentiellement subdivisés en deux grandes parties : la partie application (dénommée appli) et la partie interface (dénommée interface). Suivant chaque partie il y a des subdivisions dont voici la structure.



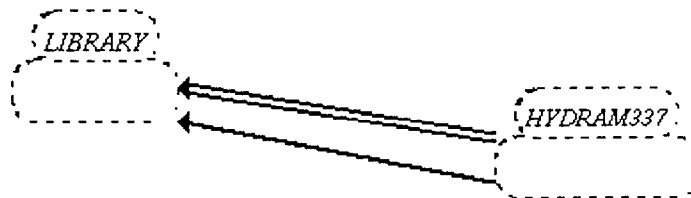




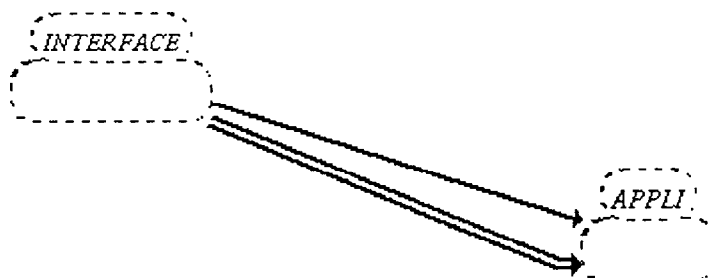
3.2. Liens d'héritage et client entre les clusters de l'application

Les liens rencontrés entre les clusters de l'application sont du type lien d'héritage et du type lien client.

Les classes qui ont été écrites dans le cadre de l'application HYDRAM sont contenues dans le cluster HYDRAM337 (337 pour signifier que c'est la version 3.3.7 d'Eiffel qui a été utilisée). Ce cluster *hérite* du cluster LIBRARY ; un cluster bibliothèque fourni par Eiffel. Il possède aussi un lien de type *client* avec ce cluster de bibliothèque d'Eiffel. Dans la suite du document les liens de type héritage seront symbolisés par un trait fléché dans le sens d'héritage fils à père. Les liens de type clients seront eux matérialisés par un double trait, fléché dans le sens client vers le fournisseur² si l'on considère qu'un cluster client utilise les clusters d'un autre qui lui est fournisseur.



Le cluster HYDRAM337 contient deux groupes de clusters : le cluster INTERFACE et le cluster APPLI. Le cluster INTERFACE possède des liens de type héritage et client avec le cluster APPLI. Entre d'autres termes le cluster INTERFACE hérite et utilise le cluster APPLI.



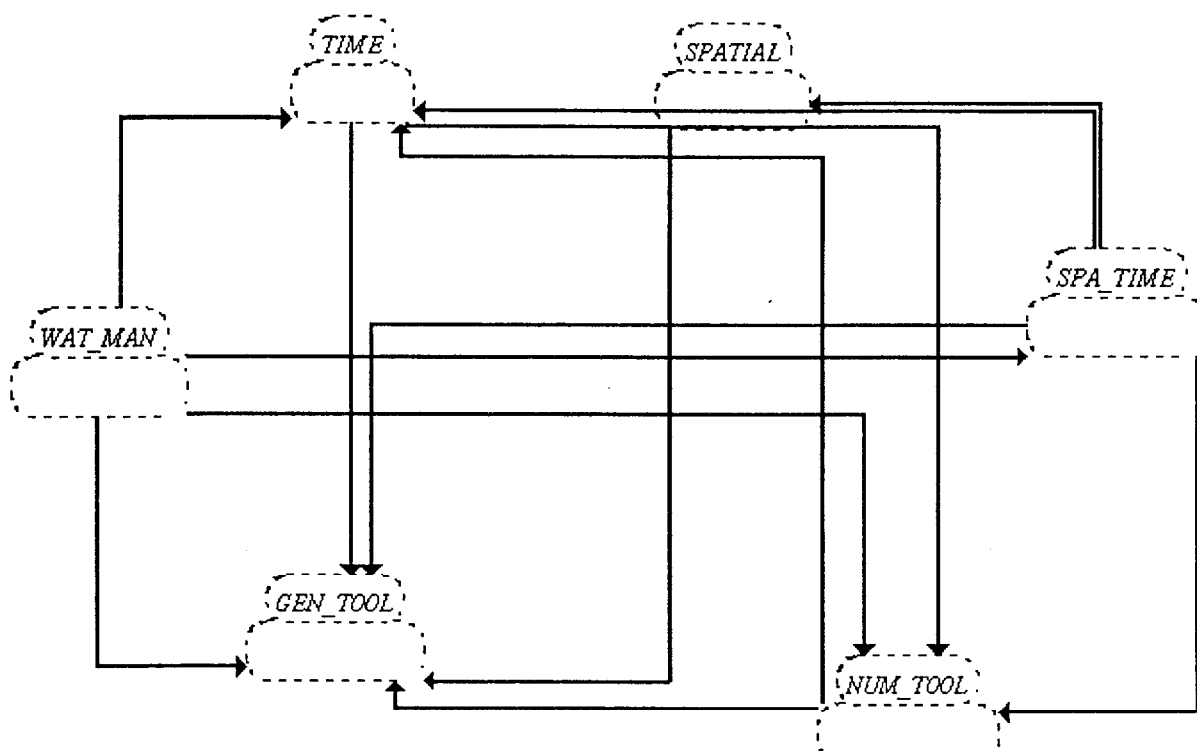
² Cette notion est à considérer avec discernement car a un autre sens dans Eiffel. Le sens utilisé ici est pour illustrer la notion de donneur et de receveur. En considérant qu'un receveur, reçoit toujours d'un donneur.

3.2.1. Liens d'héritage et client entre les clusters du cluster APPLI

3.2.1.1. Liens d'héritage entre les clusters du cluster APPLI

Comme le nom du cluster (APPLI) l'indique ; c'est ce cluster qui contient les clusters directement nécessaires et importants pour l'application HYDRAM. Ce cluster contient six autres clusters : TIME (contient les clusters directement liés à la notion de temps), SPA_TIME (contient les clusters directement liés à la notion d'espace et de temps), SPATIAL (contient les clusters directement liés à la notion d'espace), WAT_MAN (contient les clusters intervenant dans la gestion des éléments de l'application), NUM_TOOL (contient les clusters considérés comme les outils de calculs), GEN_TOOL (contient les clusters considérés comme des outils généraux de l'application).

Les liens d'héritage qui existent entre ces clusters sont représentés dans ce graphique :

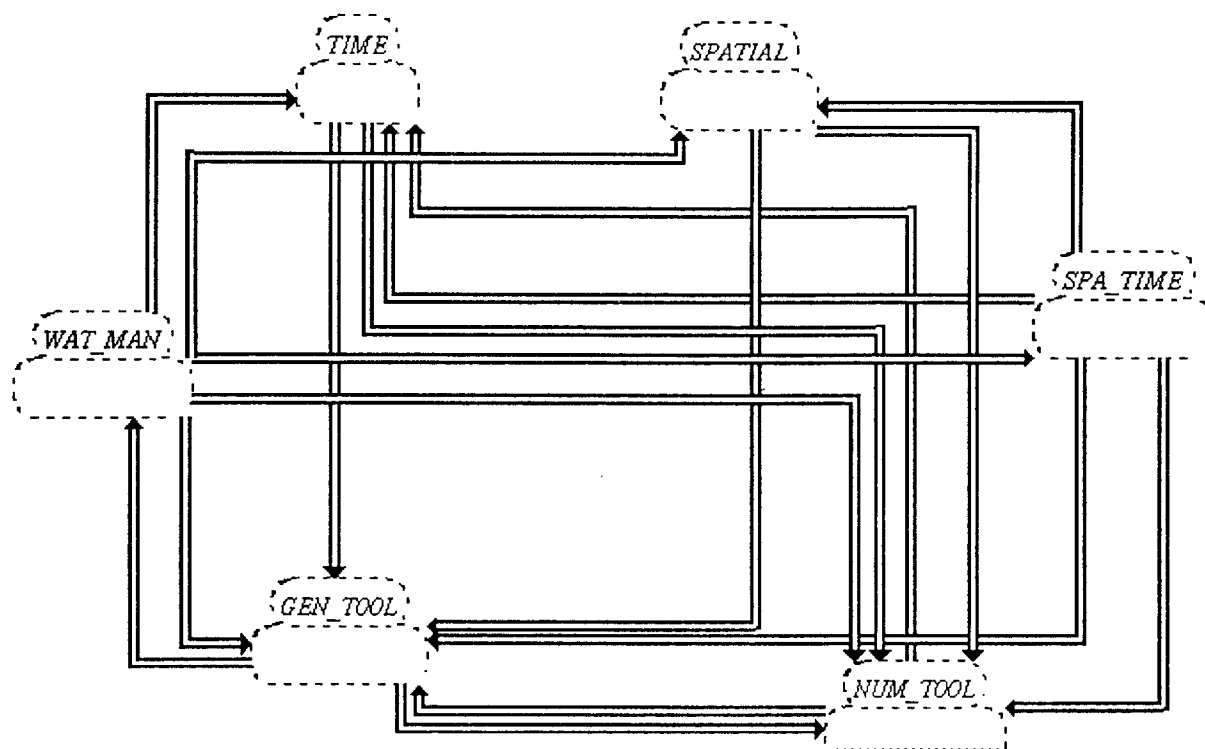


Le cluster père est GEN_TOOL ; les autres clusters fils sont WAT_MAN, TIME, SPATIAL, SPA_TIME, NUM_TOOL. Naturellement le cluster SPA_TIME hérite des clusters SPATIAL et TIME, selon le principe d'héritage multiple.

3.2.1.2. Liens client entre les clusters du cluster APPLI

Notons que dans le cluster APPLI, tous les clusters, TIME, SPATIAL, SPA_TIME, WAT_MAN sont des clients des clusters GEN_TOOL et NUM_TOOL. Cela veut dire que les clusters GEN_TOOL et NUM_TOOL sont les principaux clusters, et sont sollicités par les autres. Quant au cluster WAT_MAN ; il est client de tous les autres : il sollicite donc ces clusters.

Ce cluster APPLI est celui qui contient les clusters définissant l'application HYDRAM.



Ce cluster est très bien alimenté en liens de type d'héritage et de type client. Ce qui montre une fois de plus l'importance de celui-ci dans les échanges entre les clusters de l'application. Il est le siège même de toutes les opérations, de calculs, de définitions des éléments spatio-temporels de l'application.

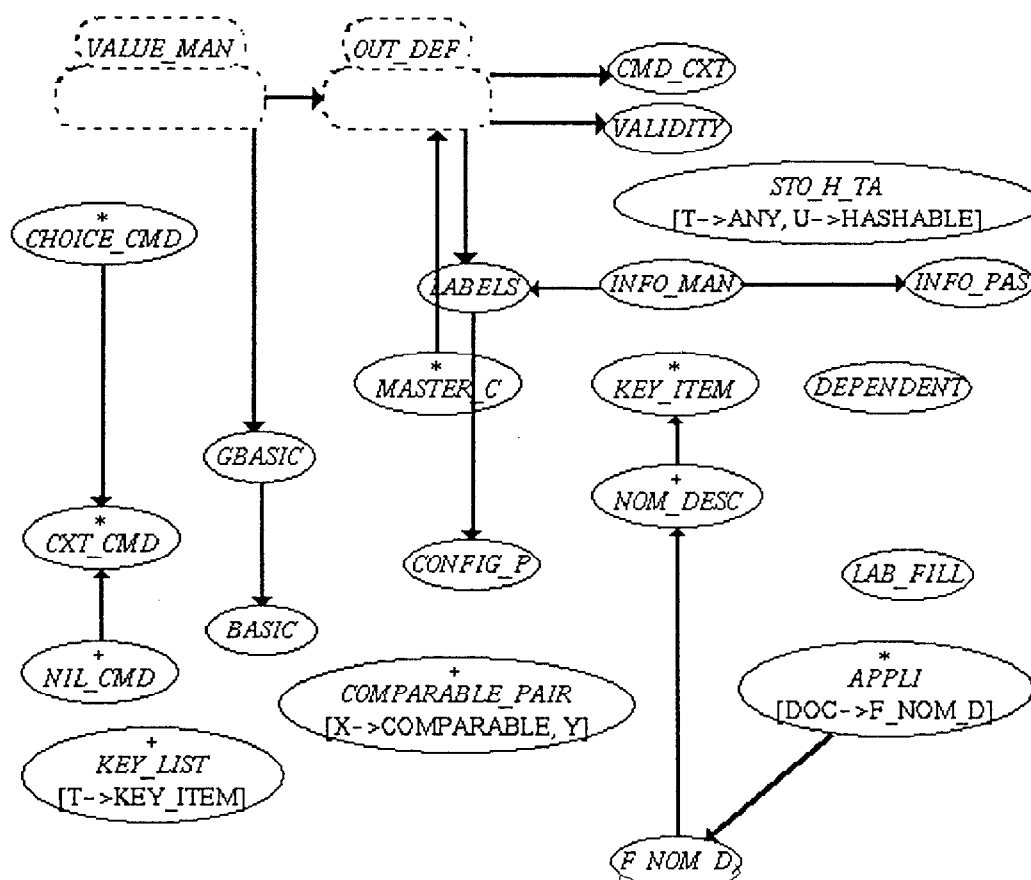
3.2.2. Liens d'héritage et client entre les clusters du cluster GEN_TOOL

3.2.2.1. Liens d'héritage entre les clusters et classes du cluster GEN_TOOL

Ce cluster GEN_TOOL contient deux autres clusters : VALUE_MAN et OUT_DEF, avec un lien d'héritage de VALUE_MAN vers OUT_DEF. Ces clusters ont des liens d'héritage avec des classes³ qui sont contenues directement dans le cluster GEN_TOOL.

Les deux clusters héritent de certaines classes, CMD_CXT, VALIDITY et LABELS pour OUT_DEF ; GBASIC pour VALUE_MAN. En fait ceci n'est qu'une représentation donnant une idée globale des liens entre les classes des clusters OUT_DEF, VALUE_MAN et les autres classes du cluster APPLI. Dire donc qu'un cluster hérite d'une classe, reste une approche globale pour dire que ce sont des classes du cluster qui héritent de la dite classe.

La notion d'héritage n'est formelle que d'une classe à une autre.



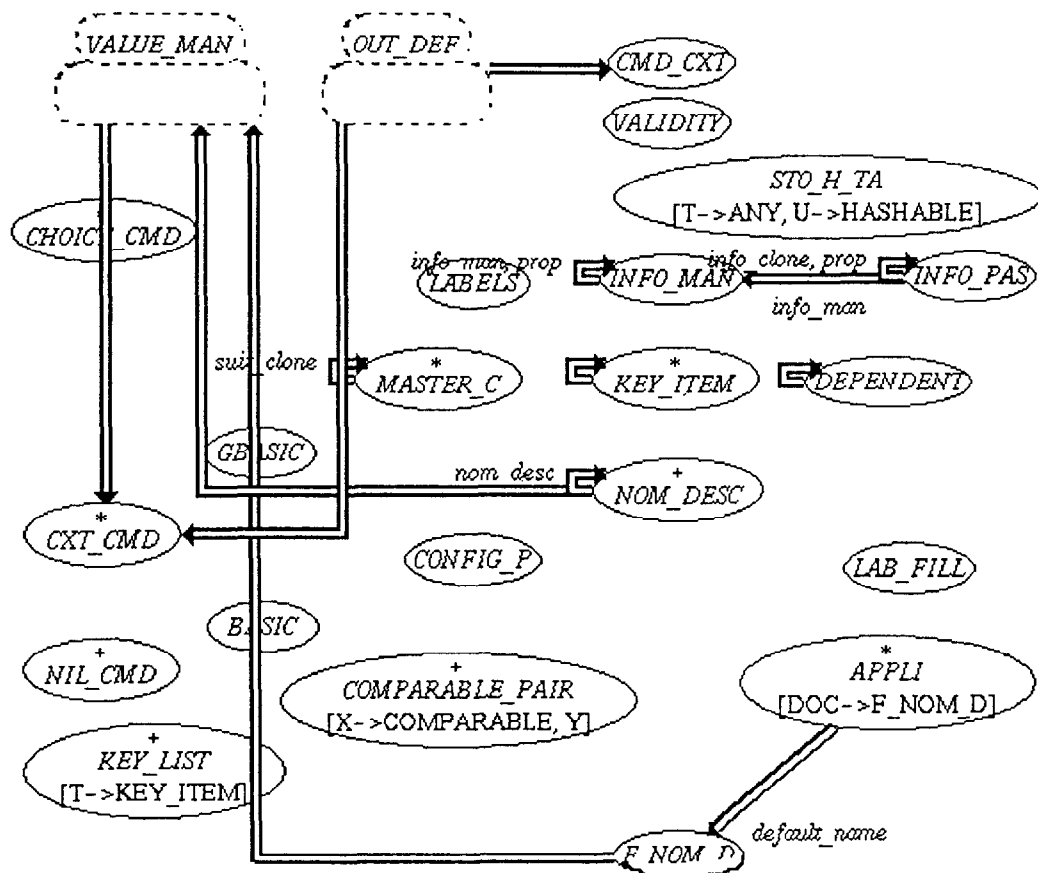
³Nous reviendrons sur ces types de liens entre classes dans un autre paragraphe.

3.2.2.2. Liens client entre les clusters et classes du cluster GEN_TOOL

Il n'y a pas de liens client entre les deux clusters VALUE_MAN et OUT_DEF. Mais ces deux clusters possèdent des liens client avec les classes du cluster GEN_TOOL.

Dire qu'un cluster est client d'une classe est aussi une approche globale pour dire que ce sont des classes du cluster qui sont clientes de la dite classe.

Il en est de même de dire qu'une classe est client d'un cluster ; la notion de lien client n'étant formelle que d'une classe à une autre.



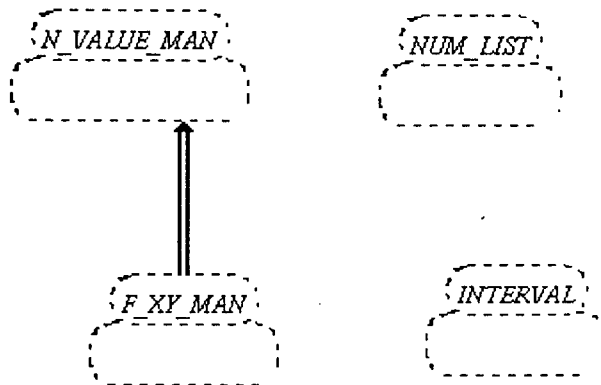
Il existe des liens client qui portent des labels dans ce cluster. Ces labels sont le plus souvent des noms de variables.

3.2.3. Liens d'héritage et client entre les clusters du cluster NUM_TOOL

Le cluster NUM_TOOL contient quatre autres clusters : N_VALUE_MAN, NUM_LIST, F_XY_MAN et INTERVAL. Il existe un lien client du cluster F_XY_MAN vers le cluster N_VALUE_MAN.

Dans ce cluster NUM_TOOL il n'existe pas de lien d'héritage entre deux quelconque clusters.

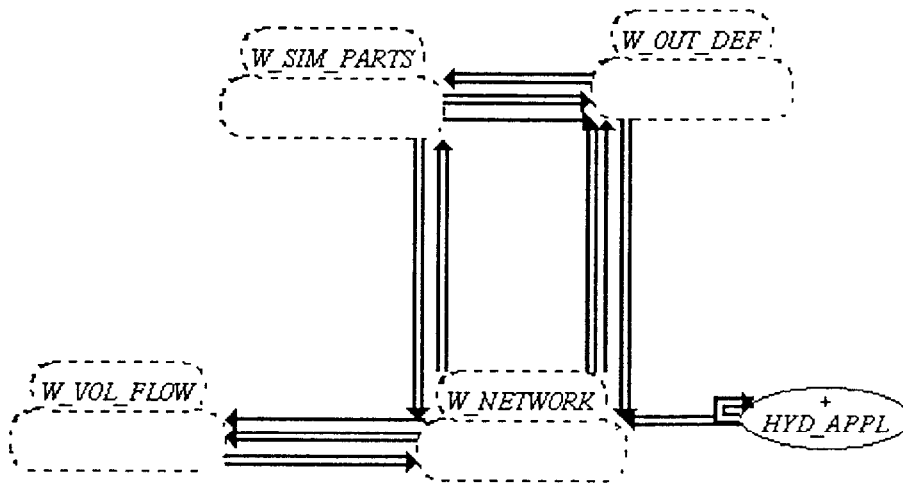
Notons que ses clusters possèdent cependant des liens d'héritage avec d'autres, mais ceux-ci étant placés dans autre un niveau de l'arborescence des clusters.



Des classes du cluster INTERVAL ne pourront pas hériter ou utiliser de celles de NUM_LIST. Il en est de même pour NUM_LIST et de N_VALUE_MAN ; pour F_XY_MAN et de INTERVAL etc...

3.2.4. Liens d'héritage et client entre les clusters du cluster WAT_MAN

Le cluster WAT_MAN contient quatre clusters qui sont : W_SIM_PARTS, W_OUT_DEF, W_VOL_FLOW et W_NETWORK. Les clusters W_SIM_PARTS et W_NETWORK héritent du cluster W_OUT_DEF. Le cluster W_NETWORK hérite du cluster W_VOL_FLOW. Nous avons des liens client dans tous les sens, entre les quatre clusters. Une classe (HYD_APPL) possède un lien client avec elle même et avec le cluster W_NETWORK.



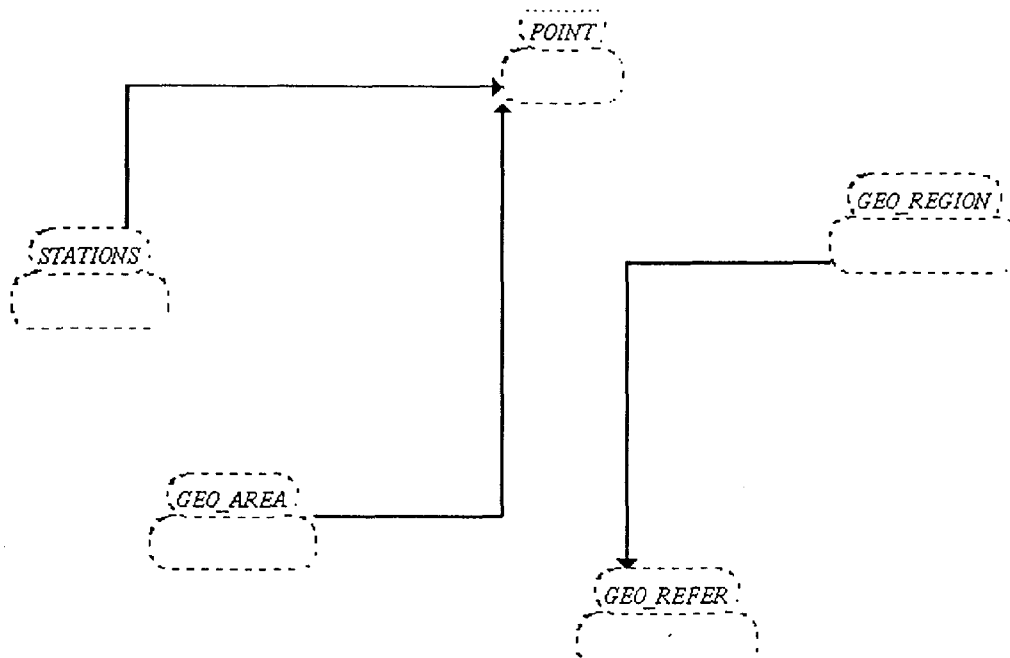
Ce cluster contient les classes et clusters, permettant de décrire et de gérer un système d'eau⁴.

⁴Un système d'eau peut être représenté par des composants subdivisés en deux grands groupes : les noeuds et les liens. Les noeuds regroupent tous les composants pouvant être localisés sans ambiguïté (une demande, un besoin, ou une jonction simple par exemple).

3.2.5. Liens d'héritage et client entre les clusters du cluster SPATIAL

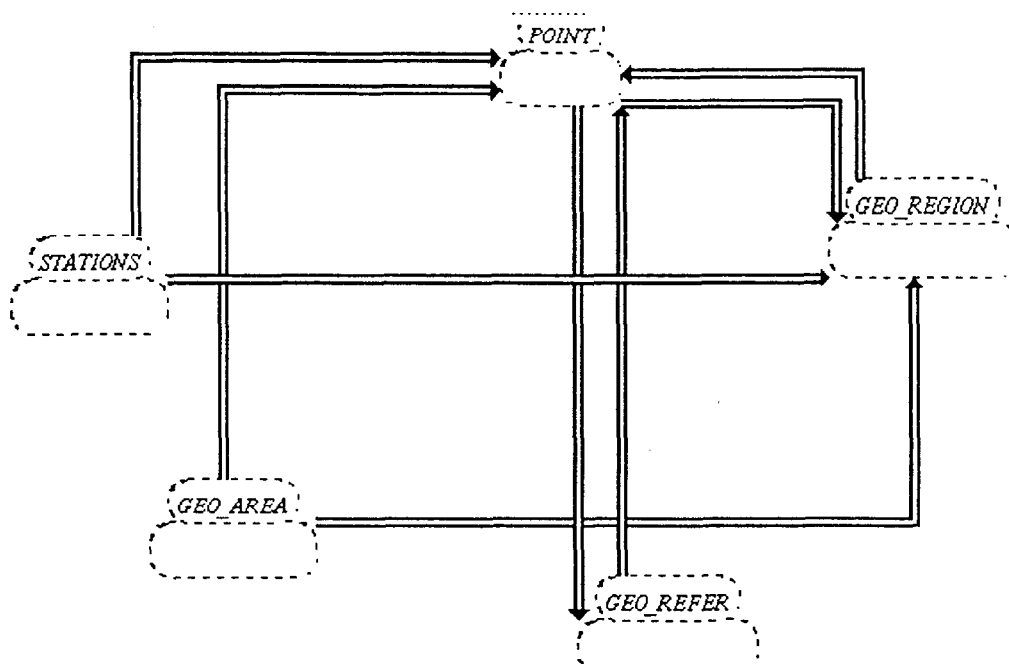
3.2.5.1. Liens d'héritage entre les clusters du cluster SPATIAL

Le cluster SPATIAL contient cinq clusters : POINT, STATIONS, GEO_REFER, GEO_AREA et GEO_REGION. Les deux clusters, GEO_AREA et STATIONS héritent du cluster POINT. Le cluster GEO_REGION hérite du cluster GEO_REFER.



3.2.5.2. Liens client entre les clusters du cluster SPATIAL

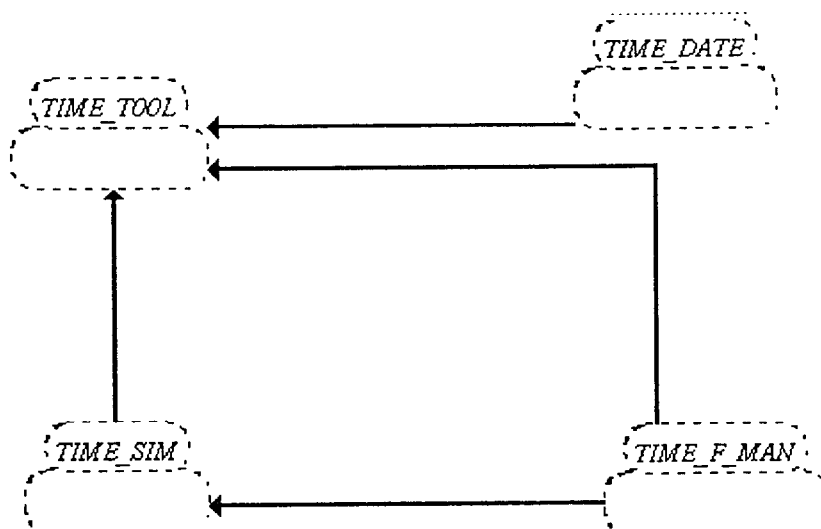
Tous les clusters sont clients de POINT. Les clusters STATIONS et GEO_AREA n'ont cependant pas de cluster client.



3.2.6. Liens d'héritage et client entre les clusters du cluster TIME

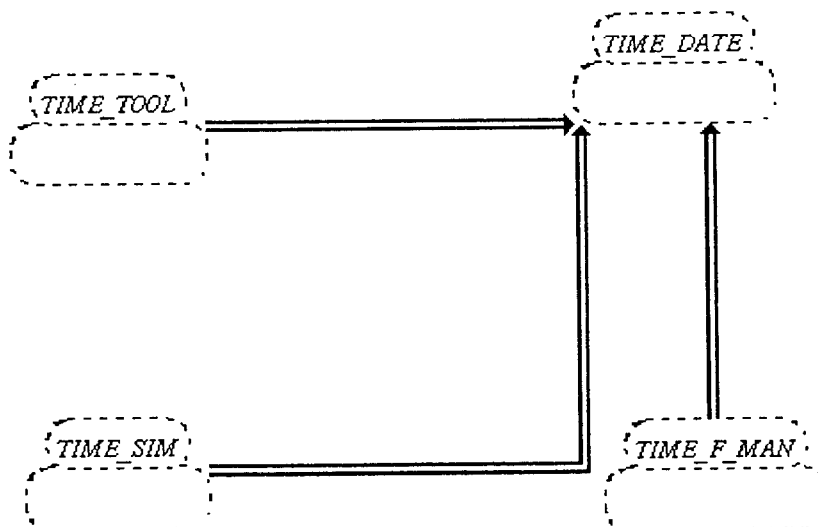
3.2.6.1. Liens d'héritage entre les clusters du cluster TIME

Tous les clusters TIME_DATE, TIME_F_MAN et TIME_SIM héritent du cluster TIME_TOOL. Le cluster TIME_F_MAN hérite des clusters TIME_SIM et TIME_TOOL, par le principe d'héritage multiple.



3.2.6.2. Liens client entre les clusters du cluster TIME

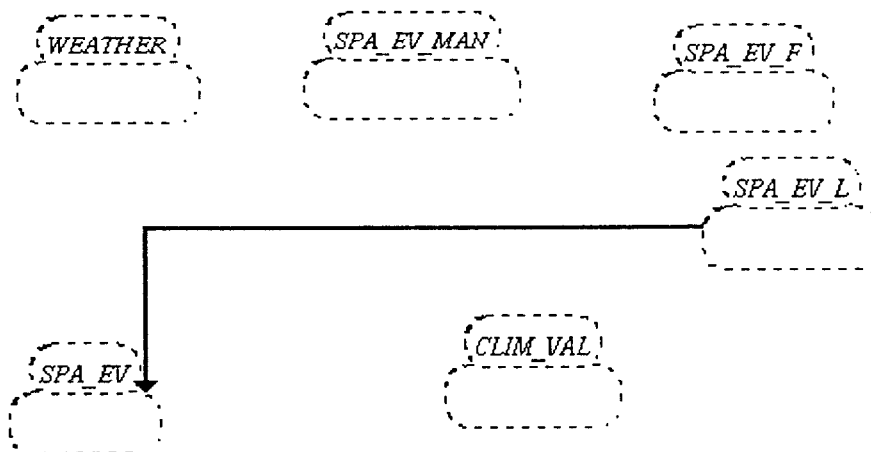
Tous les clusters TIME_TOOL, TIME_SIM et TIME_F_MAN sont clients du cluster TIME_DATE.



3.2.7. Liens d'héritage et client entre les clusters du cluster SPA_TIME

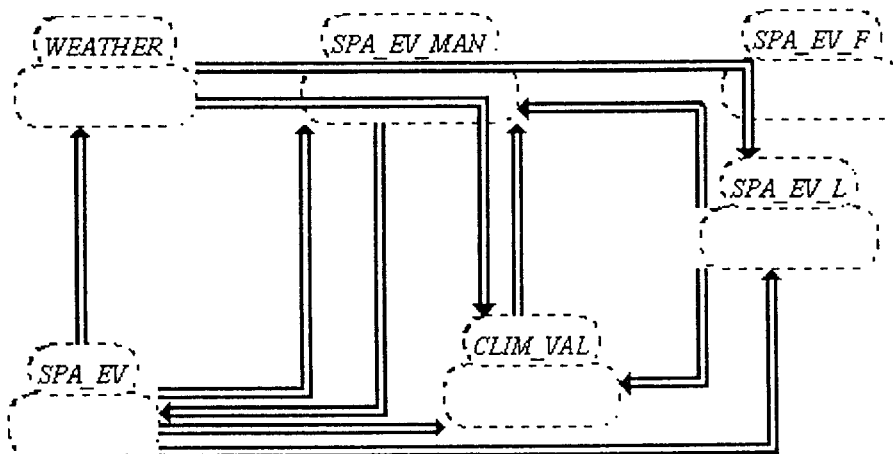
3.2.7.1. Liens d'héritage entre les clusters du cluster SPA_TIME

Le cluster SPA_TIME contient les clusters WEATHER, SPA_EV_MAN, SPA_EV_F, SPA_EV_L, CLIM_VAL et SPA_EV. Il existe un seul lien de type héritage de SPA_EV_L vers SPA_EV.



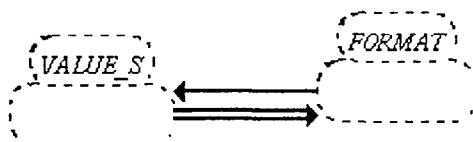
3.2.7.2. Liens client entre les clusters du cluster SPA_TIME

Seul le cluster SPA_EV_MAN ne possède pas de lien client ; soit entrant soit sortant, avec un autre cluster de SPA_TIME.



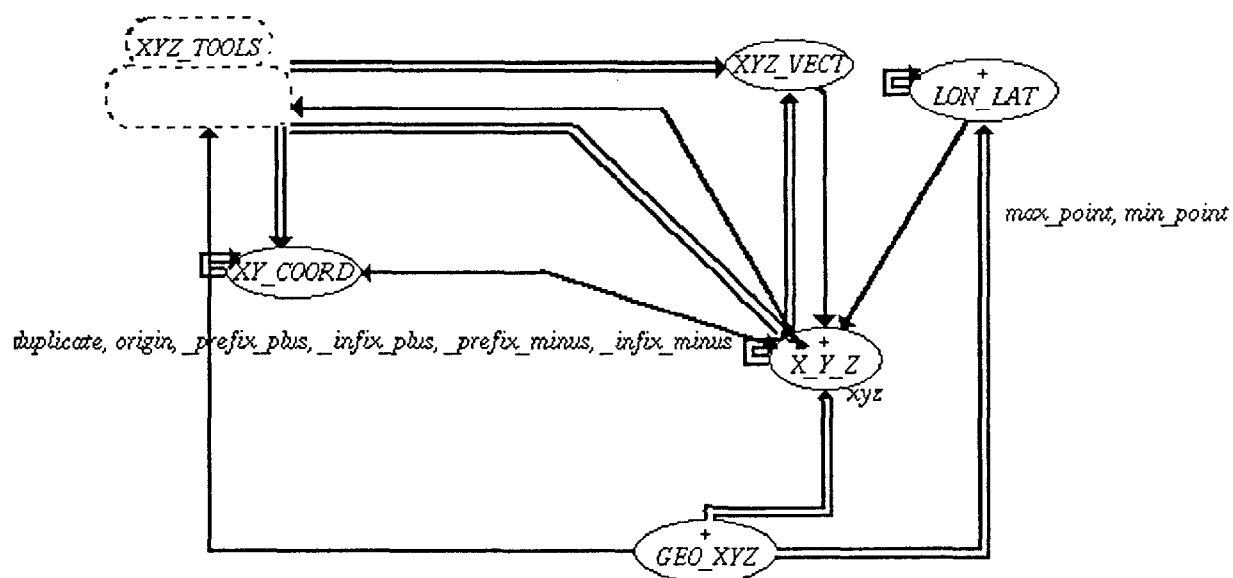
3.2.8. Liens d'héritage et client entre les clusters du cluster VALUE_MAN

Le cluster VALUE_MAN contient deux clusters, VALUE_S et FORMAT. Le cluster FORMAT hérite du cluster VALUE_S.



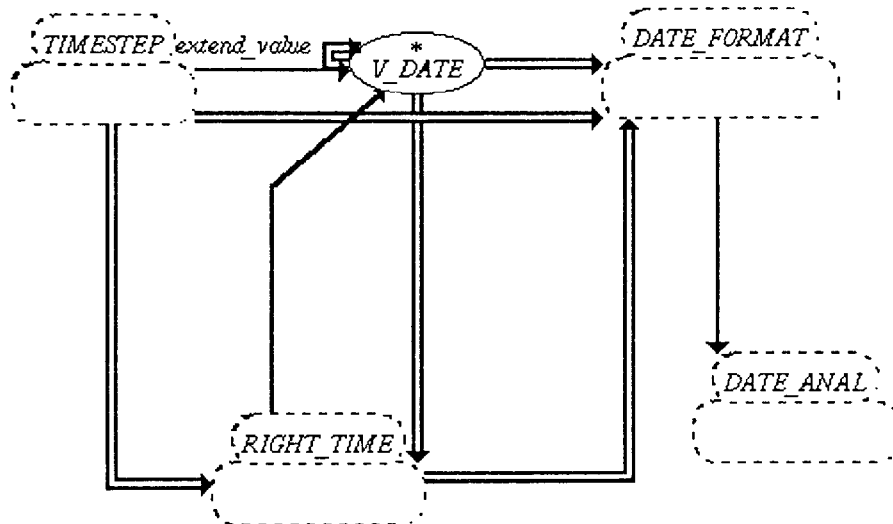
3.2.9. Liens d'héritage et client entre les clusters du cluster POINT

Le cluster POINT contient le cluster XYZ_TOOLS. Celui-ci possède des liens de type client avec les classes XYZ_VECT, XY_COORD et X_Y_Z. Les classes X_Y_Z et GEO_XYZ héritent du cluster XYZ_TOOLS. Les classes XYZ_VECT et LON_LAT héritent de la classe X_Y_Z.



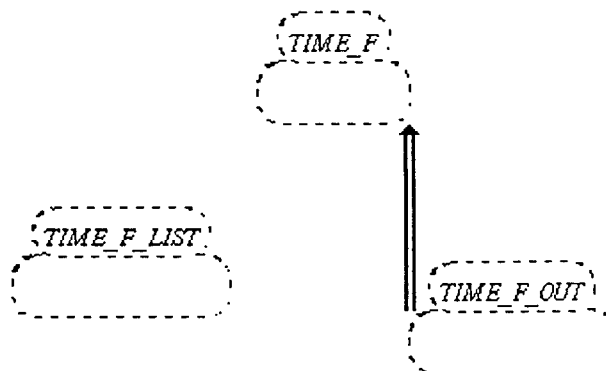
3.2.10. Liens d'héritage et client entre les clusters du cluster TIME_DATE

Le cluster TIME_DATE contient les clusters TIMESTEP, RIGHT_TIME, DATE_ANAL et DATE_FORMAT. DATE_FORMAT hérite de DATE_ANAL. Les clusters TIMESTEP et RIGHT_TIME héritent de la classe V_DATE. Les liens client vont de TIMESTEP vers RIGHT_TIME, de RIGHT_TIME vers FORMAT et de TIMESTEP vers DATE_FORMAT. La classe V_DATE possède un lien client vers DATE_FORMAT.



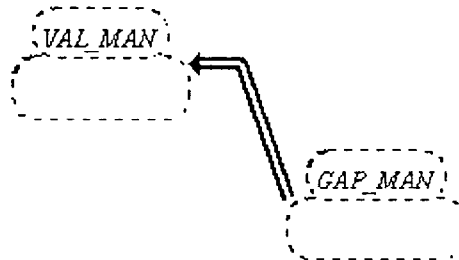
3.2.11. Liens d'héritage et client entre les clusters du cluster TIME_F_MAN

Dans ce cluster, il y a trois autres clusters : TIME_F, TIME_LIST, TIME_F_OUT. Il existe dans ce cluster un seul lien de type client de TIME_F_OUT vers TIME_F.

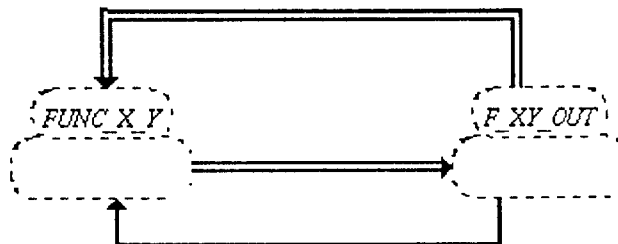


3.2.12. Liens d'héritage et client entre les clusters du cluster SPA_EV_MAN

Le cluster SPA_EV_MAN contient deux autres clusters qui sont : VAL_MAN et GAP_MAN. Il existe un seul type de lien client allant de GAP_MAN vers VAL_MAN.

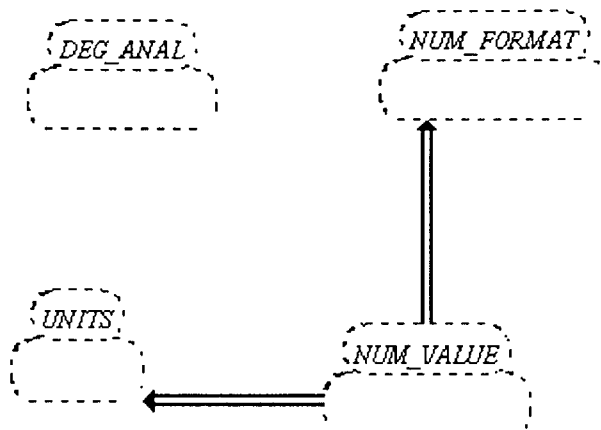
**3.2.13. Liens d'héritage et client entre les clusters du cluster F_XY_MAN**

Dans le cluster F_XY_MAN, il y a deux clusters : FUNC_X_Y et F_XY_OUT. Le cluster F_XY_OUT hérite du cluster FUNC_X_Y. Les deux clusters possèdent chacun un lien de type client de l'un vers l'autre et réciproquement.



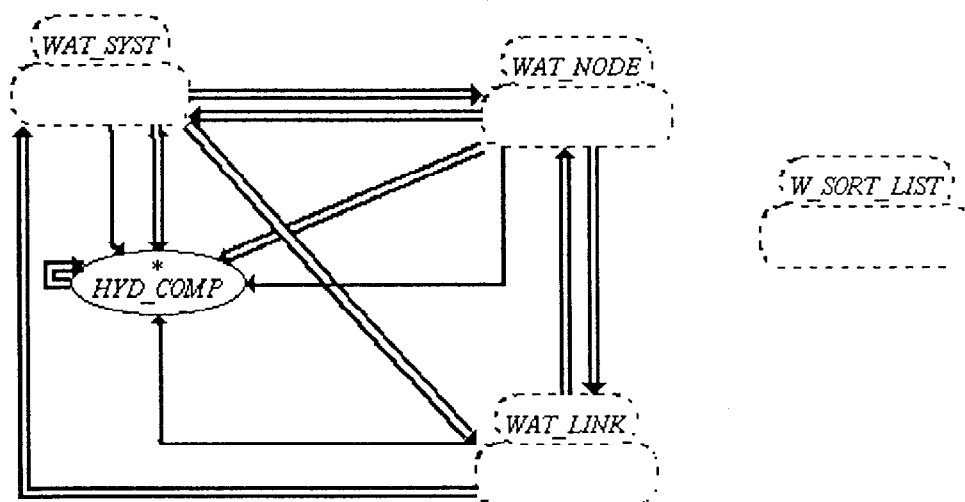
3.2.14. Liens d'héritage et client entre les clusters du cluster N_VALUE_MAN

Dans ce cluster, il existe quatre autres clusters qui sont : DEG_ANAL, NUM_FORMAT, UNITS et NUM_VALUE. Le cluster NUM_VALUE est client des clusters NUM_FORMAT et UNITS.



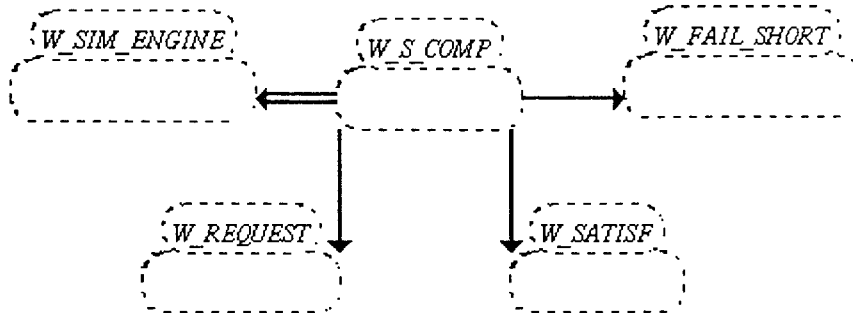
3.2.15. Liens d'héritage et client entre les clusters du cluster W_NETWORK

Celui-ci contient quatre clusters, WAT_SYST, WAT_NODE, WAT_LINK et W_SORT_LIST. Les trois premiers clusters cités possèdent chacun un lien d'héritage vers la classe HYD_COMP.



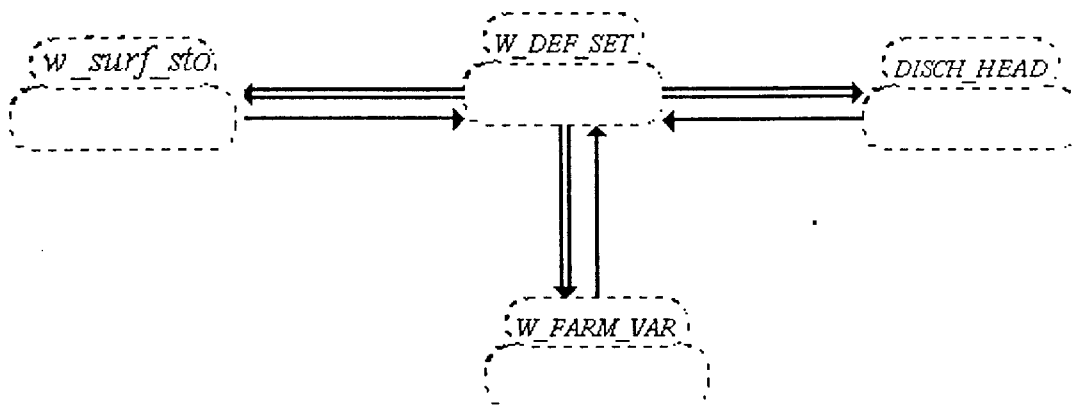
3.2.16. Liens d'héritage et client entre les clusters du cluster W_SIM_PARTS

Le cluster W_SIM_PARTS contient cinq clusters : W_SIM_ENGINE, W_S_COMP, W_FAIL_SHORT, W_REQUEST et W_SATISF. Le cluster W_S_COMP hérite des clusters W_REQUEST, W_FAIL_SHORT et W_SATISF. Seul le cluster W_S_COMP possède un lien de type client vers W_SIM_ENGINE.



3.2.17. Liens d'héritage et client entre les clusters du cluster W_OUT_DEF

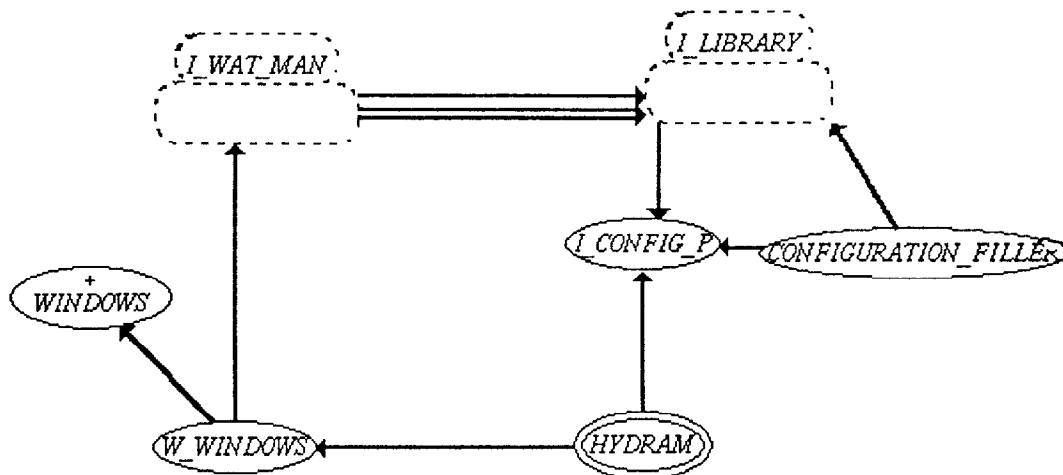
Ce cluster W_OUT_DEF contient les clusters W_DEF_SET, W_SURF_STO, DISCH_HEAD et W_FARM_VAR. Les trois derniers clusters possèdent des liens de type héritage vers le cluster W_SET_DEF. Le cluster W_SET_DEF est le seul à avoir des liens de type client vers les autres clusters.



3.3. Liens d'héritage et client entre les clusters de l'interface

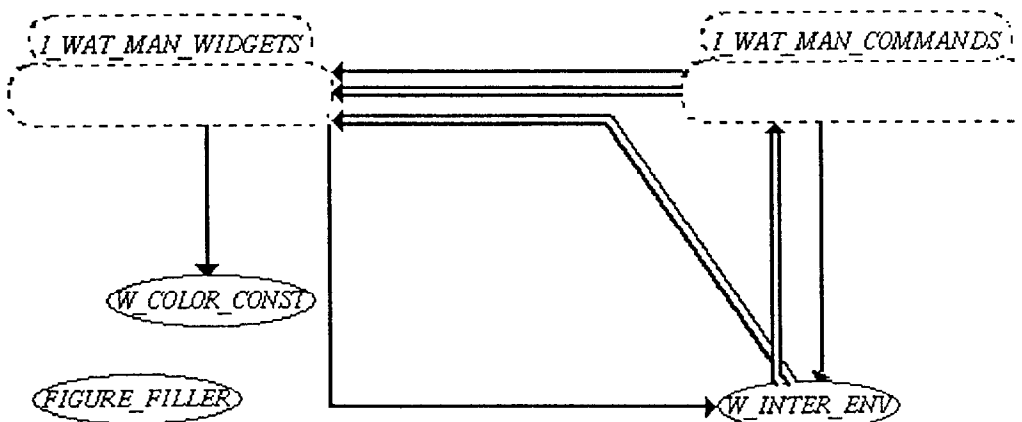
3.3.1. Liens d'héritage et client entre les clusters du cluster INTERFACE

Ce cluster contient deux autres clusters : I_WAT_MAN et I_LIBRARY. Le cluster I_WAT_MAN hérite du cluster I_LIBRARY. La classe racine du système, HYDRAM hérite des classes I_CONFIG_P et W_WINDOWS. La classe W_WINDOWS hérite de la classe WINDOWS et du cluster I_WAT_MAN. Le cluster I_LIBRARY hérite de la classe I_CONFIG_P. Quand à la classe CONFIGURATION_FILLER elle hérite de la classe I_CONFIG_P et du cluster I_LIBRARY. Le cluster I_WAT_MAN est client du cluster I_LIBRARY.



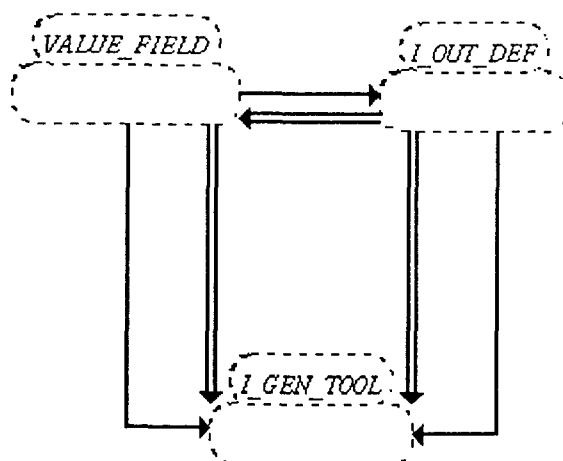
3.3.2. Liens d'héritage et client entre les clusters du cluster I_WAT_MAN

Celui-ci contient deux clusters, I_WAT_MAN_WIDGETS et I_WAT_MAN_COMMANDS. Le cluster I_WAT_MAN_COMMANDS hérite du cluster I_WAT_MAN_WIDGETS et de la classe W_INTER_ENV. Le cluster I_WAT_MAN_COMMANDS hérite des classes W_COLOR_CONST et W_INTER_ENV. La classe W_INTER_ENV est client des clusters I_WAT_MAN_COMMANDS et I_WAT_MAN_WIDGETS.



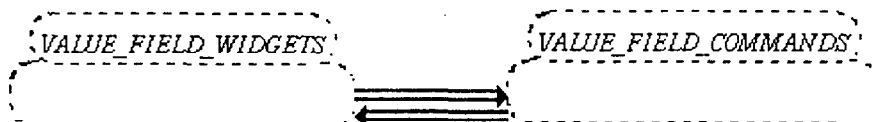
3.3.3. Liens d'héritage et client entre les clusters du cluster I_LIBRARY

Le cluster I_LIBRARY contient les clusters VALUE_FIELD, I_OUT_DEF et I_GEN_TOOL. Les clusters VALUE_FIELD et I_OUT_DEF héritent mais sont aussi des clients du cluster I_GEN_TOOL. Un lien de type héritage va de VALUE_FIELD vers I_OUT_DEF ; un autre de type client va de I_OUT_DEF vers VALUE_FIELD.



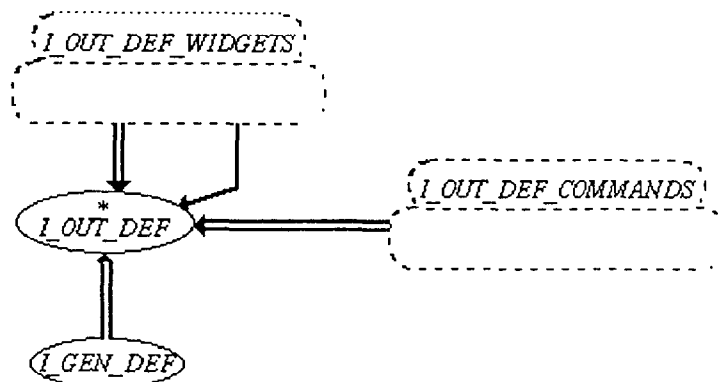
3.3.4. Liens d'héritage et client entre les clusters du cluster VALUE_FIELD

Nous avons deux clusters : VALUE_FIELD_WIDGETS et VALUE_FIELD_COMMANDS ayant chacun un lien de type client de l'un vers l'autre.



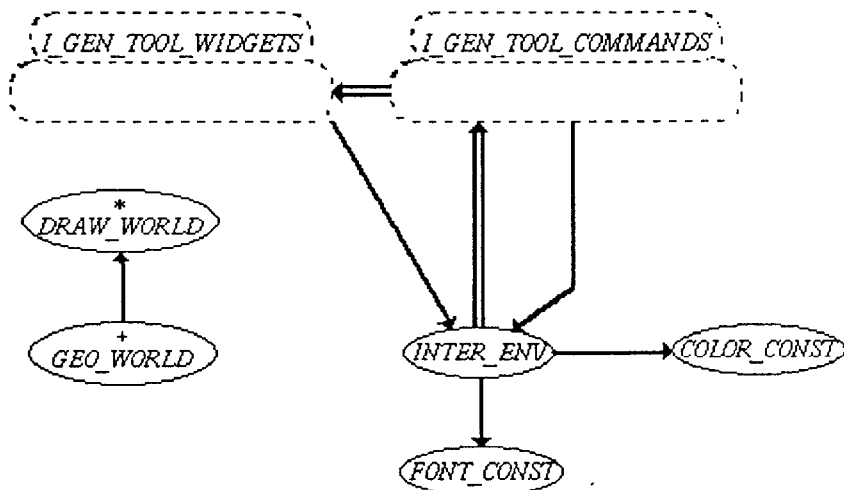
3.3.5 Liens d'héritage et client entre les clusters du cluster I_OUT_DEF

Le cluster I_OUT_DEF_WIDGETS hérite de la classe I_OUT_DEF. Les clusters I_OUT_DEF_COMMANDS, I_OUT_DEF_WIDGETS et la classe I_GEN_DEF sont clients de la classe I_OUT_DEF.



3.3.6. Liens d'héritage et client entre les clusters du cluster I_GEN_TOOL

Le cluster I_GEN_TOOL contient deux clusters : I_GEN_TOOL_WIDGETS et I_GEN_TOOL_COMMANDS. Le cluster I_GEN_TOOL_COMMANDS est client du cluster I_GEN_TOOL_WIDGETS. Ces deux clusters héritent de la classe INTER_ENV ; qui elle à son tour hérite des classes FONT_CONST et COLOR_CONST. La classe INTER_ENV est client du cluster I_GEN_TOOL_COMMANDS. La classe GEO_WORLD hérite de la classe DRAW_WORLD.



4. Liens d'héritage et client entre classes

Notre démarche dans ce paragraphe, sera de considérer un à un les clusters vus dans le paragraphe précédent, de passer en revue les différentes classes qui s'y trouvent.

Il sera question d'indiquer les différents types de liens entre les classes d'un cluster donné, par une représentation graphique du contenu du cluster.

Comme ceci l'a été au paragraphe précédent nous allons considérer deux grandes parties de classes : les classes de l'application, et celles de l'interface.

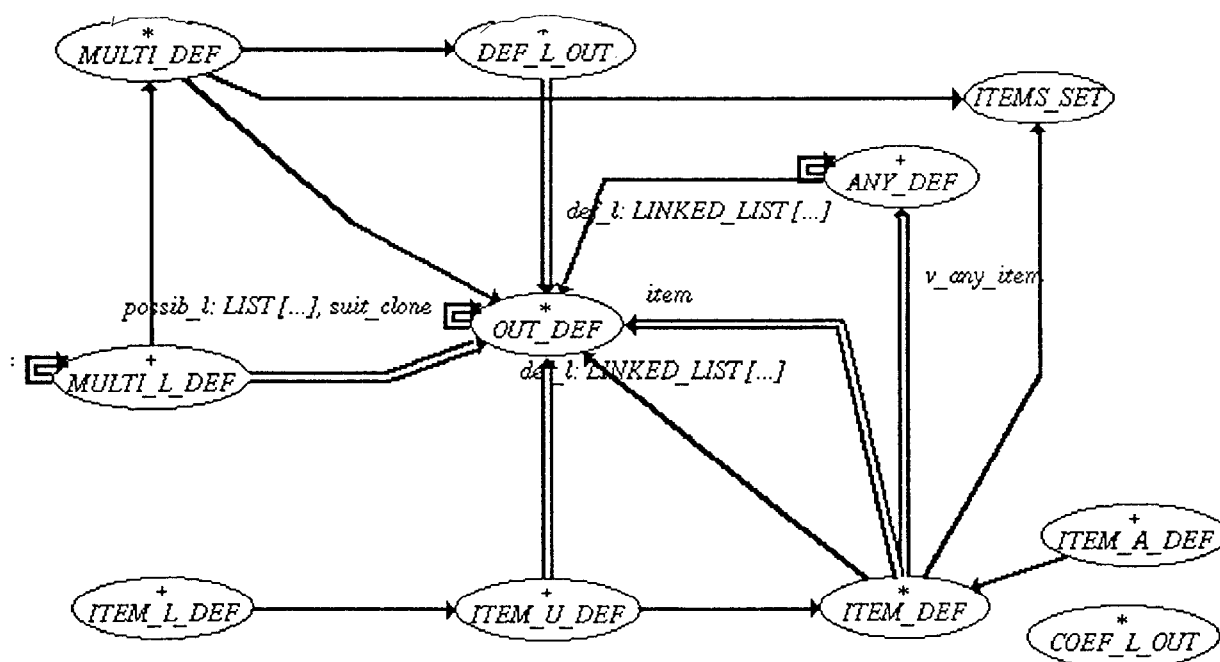
Il sera exclu dans la présentation de ce paragraphe, les classes ayant directement des liens avec des clusters, et qui ont été abordées dans le paragraphe précédent.

Les clusters "terminaux"⁵ ne contenant qu'une seule classe ne seront pas représentés graphiquement, car il y a aucun intérêt du fait de l'absence de liens d'héritage ou client.

4.1. Liens d'héritage et client entre les classes de l'application

4.1.1. Liens d'héritage et client entre les classes du cluster OUT_DEF

La classe de base est OUT_DEF (OUT_DEFINITIONS - définitions et sortie), c'est une classe différée. Toutes les autres classes du cluster héritent directement ou indirectement de cette classe.



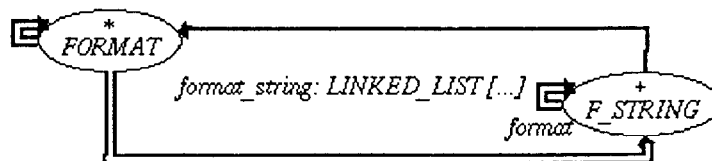
⁵ Des clusters qui ne contiennent plus d'autres clusters, donc ils ne contiennent que des classes uniquement.

La classe de base de ce cluster est VALUE_S (VALUE_String - valeur et correspondance avec une chaîne de caractère), c'est une classe différée. Toutes les autres classes du cluster héritent directement ou indirectement de cette classe.

The diagram is a semantic network with the following nodes and relationships:

- Nodes:**
 - $UNIX_PATH^+$
 - DOC_NAME^+
 - VD_STRING^*
 - $PATH_NAMEEE^*$
 - $V_C_STRING^+$
 - V_STRING^+
 - $S11_99^+$
 - DOS_PATH^+
 - EXT_NAME^+
 - E_REAL^+
 - $E_LIM_NUMB^*$
 - $VALUE_E^*$
 - $V_BOOL_REF^+$
 - $VD_BOOLEAN^*$
 - $VALUE_S^{*model}$
- Relationships (Arrows):**
 - $UNIX_PATH^+ \rightarrow VD_STRING^*$
 - $DOC_NAME^+ \rightarrow VD_STRING^*$
 - $VD_STRING^* \rightarrow PATH_NAMEEE^*$
 - $V_C_STRING^+ \rightarrow V_STRING^+$
 - $V_C_STRING^+ \rightarrow VD_STRING^*$ (double arrow)
 - $V_C_STRING^+ \rightarrow VALUE_S^{*model}$ (double arrow)
 - $V_STRING^+ \rightarrow VD_STRING^*$
 - $S11_99^+ \rightarrow VD_STRING^*$
 - $DOS_PATH^+ \rightarrow PATH_NAMEEE^*$
 - $EXT_NAME^+ \rightarrow VD_STRING^*$
 - $E_REAL^+ \rightarrow E_LIM_NUMB^*$
 - $E_LIM_NUMB^* \rightarrow VALUE_E^*$
 - $VALUE_E^* \rightarrow VALUE_S^{*model}$ (double arrow)
 - $V_BOOL_REF^+ \rightarrow VD_BOOLEAN^*$
 - $VD_BOOLEAN^* \rightarrow VALUE_S^{*model}$ (double arrow)
 - $VD_STRING^* \xleftrightarrow{possib_c_LIST [...]} VALUE_S^{*model}$
 - $VALUE_S^{*model} \xleftrightarrow{extend_value} VD_BOOLEAN^*$
 - $VD_BOOLEAN^* \xleftrightarrow{extend_value} VALUE_S^{*model}$
- Special Features:**
 - A double arrow connects VD_STRING^* and $VALUE_S^{*model}$.
 - A small square icon with a right-pointing arrow is located near the VD_STRING^* node.

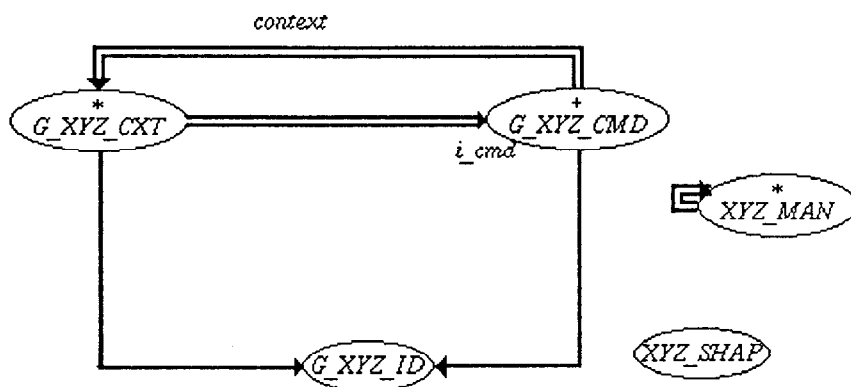
La classe de base est `FORMAT`. Une classe effective `F_STRING`, hérite de la classe `FORMAT`. La classe `FORMAT` utilise `F_STRING` à travers une variable ou label appelé *format*.



4.1.4. Liens d'héritage et client entre les classes du cluster XYZ_TOOLS

Les classes G_XYZ_CMD et G_XYZ_CXT héritent de la classe G_XYZ_ID. Elles sont clientes l'une de l'autre et réciproquement. La classe différée XYZ_MAN est cliente d'elle même.

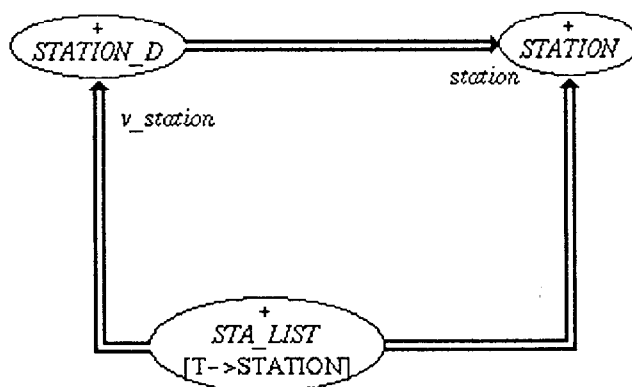
Les classes de ce cluster offrent des outils pour la gestion de l'espace à trois dimensions.



4.1.5. Liens d'héritage et client entre les classes du cluster STATIONS

Les liens qui existent entre les classes du cluster STATIONS sont uniquement des liens de type client. La classe STA_LIST[T→STATION] est cliente des classes STATION et STATION_D. La classe STATION_D est cliente de la classe STATION.

Les classes de ce cluster STATION permettent de définir une station, une liste de stations.



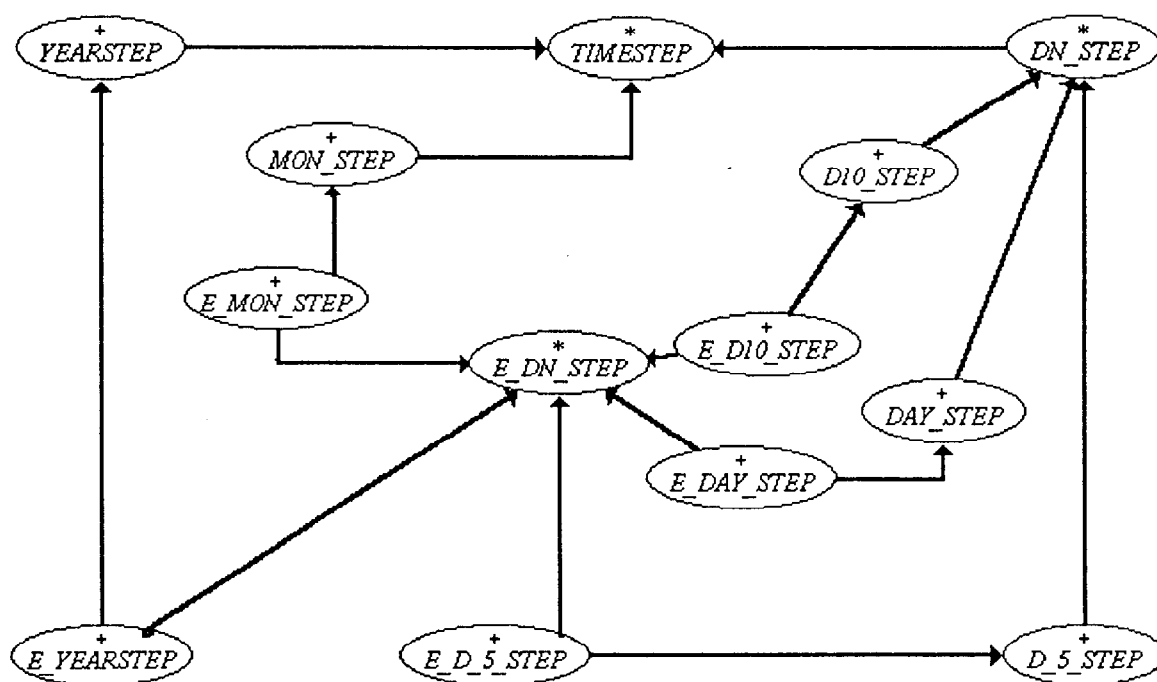
4.1.6. Liens d'héritage et client entre les classes du cluster TIMESTEP

4.1.6.1. Liens d'héritage entre les classes du cluster TIMESTEP

La classe de base dans ce cluster est TIMESTEP. Toutes les autres classes du cluster héritent directement ou indirectement de cette classe différée. Les classes qui héritent directement de TIMESTEP sont : DN_STEP, MON_STEP et YEARSTEP.

Ce cluster contient les classes décrivant la notion de temps (sa partie différentielle ou pas de temps) et qui sont utilisées dans toute l'application d'HYDRAM.

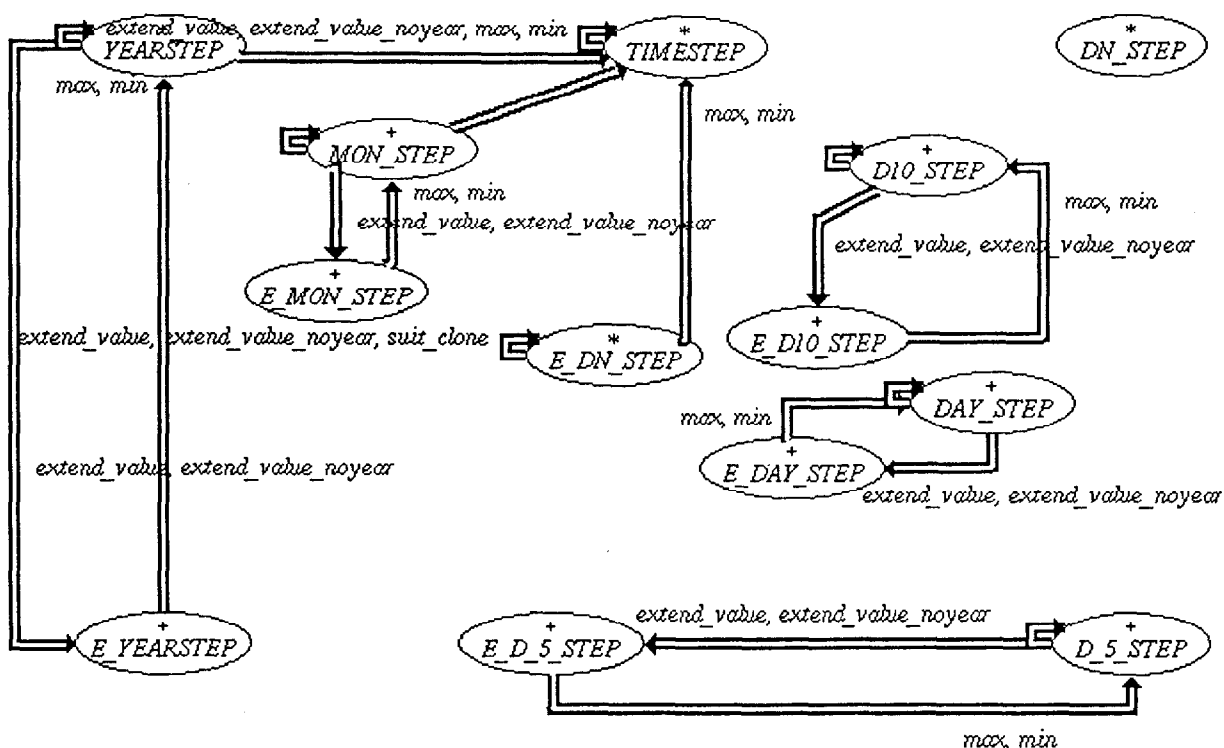
Les pas de temps utilisés au sens exhaustif sont : le journalier, le pentadaire, le décadaire, le mensuel et l'annuel (pour les moyennes interannuelles des données).



4.1.6.2. Liens client entre les classes du cluster TIMESTEP

Dans ce cluster TIMESTEP, les liens client entre classes sont nombreux.

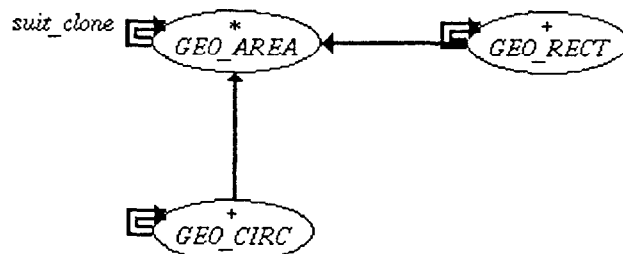
Les classes YEARSTEP, MON_STEP et E_DN_STEP sont clientes de la classe différée TIMESTEP. Les couples de classes : YEARSTEP et E_YEARSTEP, D10_STEP et E_D10_STEP, DAY_STEP et E_DAY_STEP, MON_STEP et E_MON_STEP, D_5_STEP et E_D_5_STEP forment des classes clientes l'une de l'autre (par couple de classes) et réciproquement.



On remarquera que les liens (de type client) du cluster portent des labels. Ces labels sont pour la plupart des noms de "routines".

4.1.7. Liens d'héritage et client entre les classes du cluster GEO_AREA

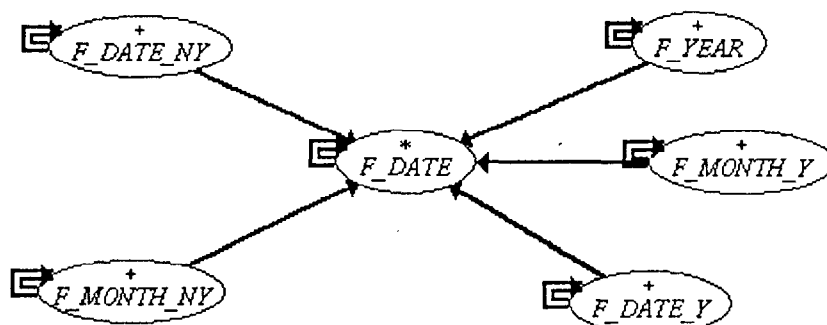
La classe de base du cluster GEO_AREA est GEO_AREA. Les classes GEO_RECT et GEO_CIRC héritent de la classe GEO_AREA. Les classes de ce cluster permettent de définir et de gérer des zones géographiques, dans une représentation schématique d'un périmètre d'irrigation par exemple.



4.1.8. Liens d'héritage et client entre les classes du cluster DATE_FORMAT

Dans le cluster DATE_FORMAT, les classes, F_YEAR, F_MONTH_Y, F_DATE_Y, F_MONTH_NY et F_DATE_NY héritent de la classe F_DATE.

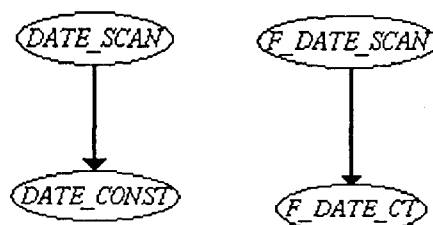
Les classes du cluster permettent de prendre en compte et de gérer les formats des dates.



4.1.9. Liens d'héritage et client entre les classes du cluster DATE_ANAL

La classe DATE_SCAN hérite de la classe DATE_CONST. Aussi, la classe F_DATE_SCAN hérite de la classe F_DATE_CT.

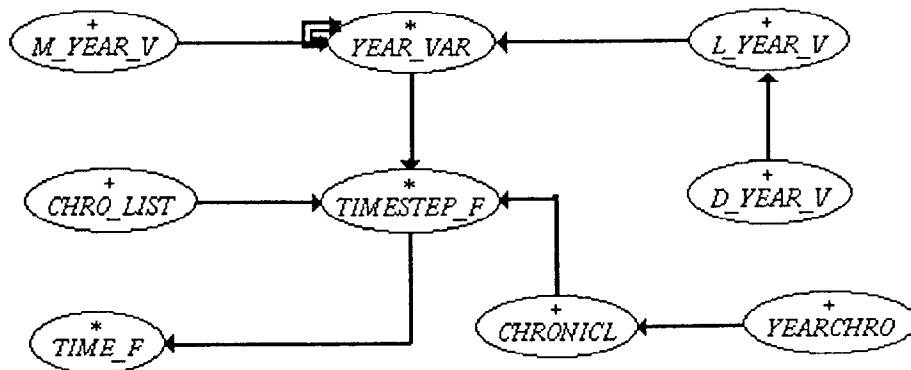
Les classes de ce cluster permettent d'analyser les dates intervenant dans l'application HYDRAM.



4.1.10. Liens d'héritage et client entre les classes du cluster TIME_F

Les classes CHRO_LIST, YEAR_VAR et CHRONICL héritent de la classe TIMESTEP_F. Les classes L_YEAR et M_YEAR_V héritent de la classe YEAR_VAR.

Les classes de ce cluster permettent de définir des fonctions bornées associant une valeur à une date ou un pas de temps.

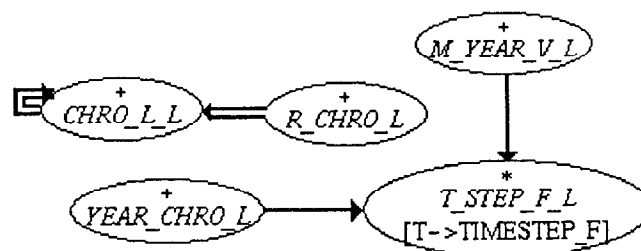


4.1.11. Liens d'héritage et client entre les classes du cluster TIME_F_LIST

Les classes M_YEAR_V_L et YEAR_CHRO_L héritent de la classe T_STEP_F_L [T→TIMESTEP_F].

La classe R_CHRO_L est cliente de la classe CHRO_L_L.

Les classes de ce cluster, permettent de prendre en compte des listes de pas de temps et des valeurs chronologiques.



4.1.11. Liens d'héritage et client entre les classes du cluster TIME_F_OUT

Le classe STEP_F_OUT hérite de la classe TIME_F_OUT.

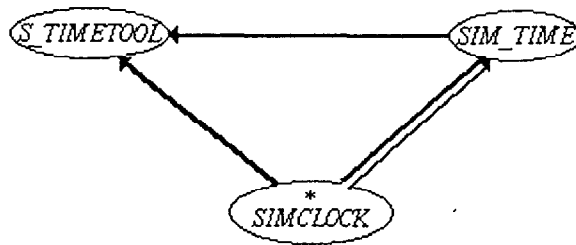
Les deux classes interviennent dans le filtre de sortie de fonction de pas de temps à réel et de réel à réel.



4.1.12. Liens d'héritage et client entre les classes du cluster TIME_SIM

Les classes SIM_TIME et SIMCLOCK héritent de la classe S_TIMETOOL. La classe SIMCLOCK est cliente de la classe SIM_TIME.

Les classes de TIME_SIM prennent en compte les outils nécessaires (en ce qui concerne le temps) pour la simulation.

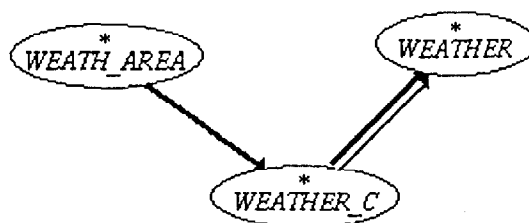


4.1.13. Liens d'héritage et client entre les classes du cluster WEATHER

La classe WEATH_AREA hérite de la classe WEATHER_C. La classe WEATHER_C est cliente de la classe WEATHER.

Toutes les classes de ce cluster sont différées. Elles ne sont pas effectives.

Ces classes interviennent dans la définition des événements (données) spatio-temporels.



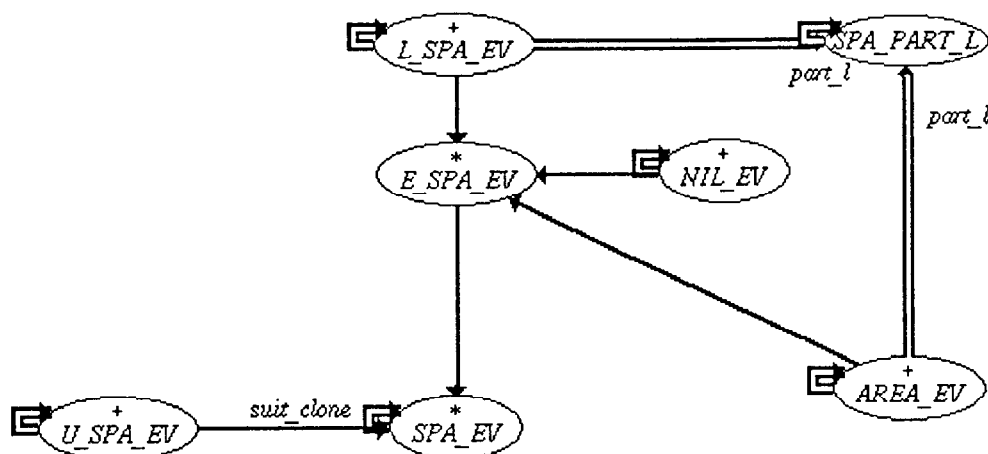
4.1.14. Liens d'héritage et client entre les classes du cluster SPA_EV

La classe de base de ce cluster est SPA_EV. Les autres classes du cluster héritent directement ou indirectement de cette classe SPA_EV.

Les classes L_SPA_EV, NIL_EV et AREA_EV héritent de la classe E_SPA_EV.

Les classes L_SPA_EV et AREA_EV sont clientes de la classe SPA_PART_L.

Ces classes permettent de prendre en compte les événements spatio-temporels.

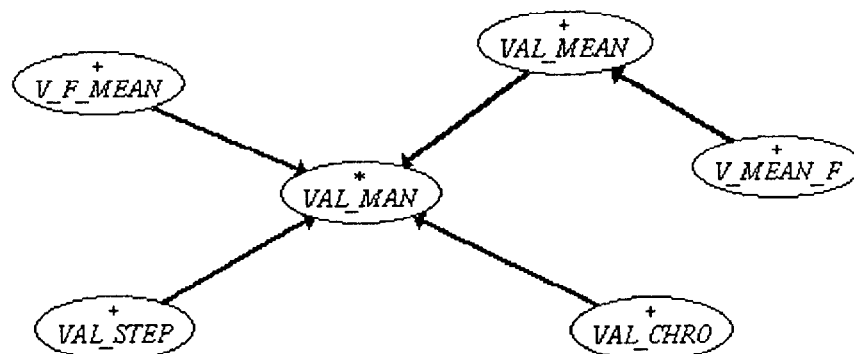


4.1.15. Liens d'héritage et client entre les classes du cluster VAL_MAN

Le cluster VAL_MAN contient les classes : VAL_MEAN, V_MEAN_F, VAL_CHRO, VAL_MAN, V_F_MEAN et VAL_STEP.

Les classes VAL_MEAN, VAL_CHRO, V_F_MEAN et VAL_STEP héritent de la classe VAL_MAN.

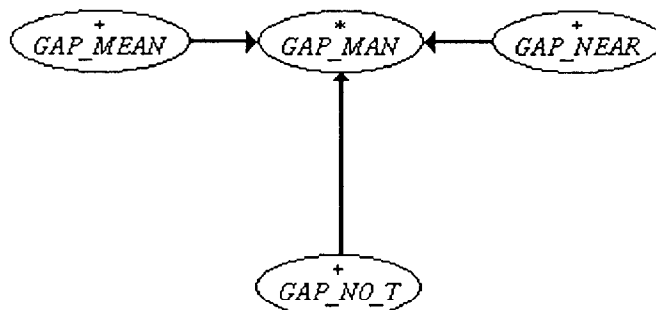
Les classes de cluster permettent de gérer les données d'un horizon prévisionnel.



4.1.16. Liens d'héritage et client entre les classes du cluster GAP_MAN

Les classes GAP_MEAN, GAP_NEAR et GAP_NO_T héritent de la classe GAP_MAN.

Les classes de ce cluster permettent de gérer les lacunes dans les données hydro-météorologiques.

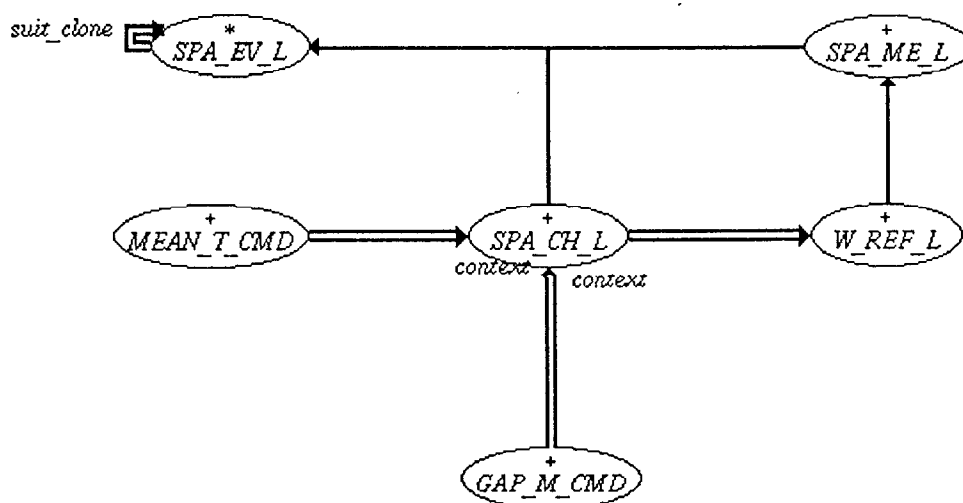


4.1.17. Liens d'héritage et client entre les classes du cluster SPA_EV_L

Les classes SPA_CH_L et SPA_ME_L héritent de la classe SPA_EV_L.

Les classes MEAN_T_CMD, GAP_M_CMD et W_REF_F sont clientes de la classe SPA_CH_L.

Les classes de ce cluster permettent de prendre en compte les listes des événements spatiaux.



4.1.18. Liens d'héritage et client entre les classes du cluster CLIM_VAI

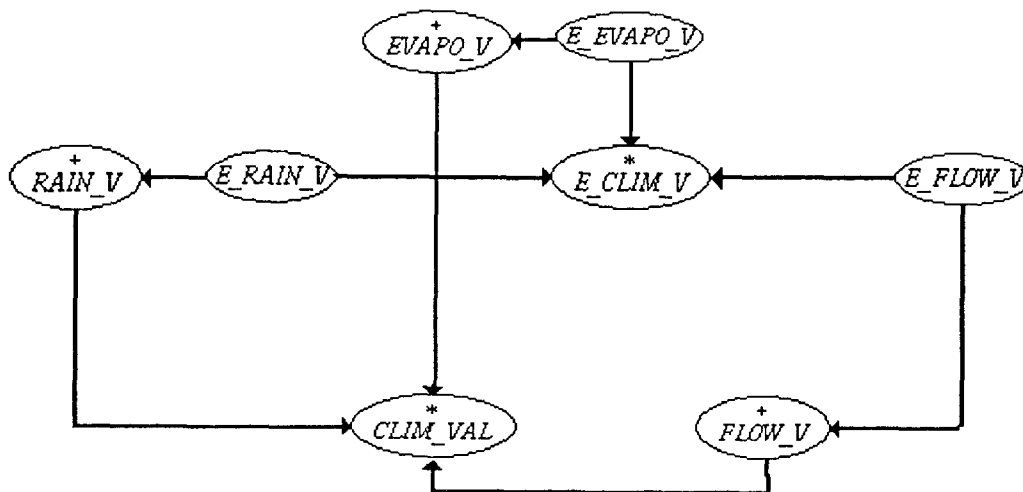
4.1.18.1. Liens d'héritage entre les classes du cluster CLIM_VAI

Les classes RAIN_V, FLOW_V et EVAPO_V héritent de la classe CLIM_VAL.

E_RAIN_V, E_FLOW_V et E_EVAPO_V héritent de la classe E_CLIM_V.

La classe E_RAIN_V hérite des classes RAIN_V et E_CLIM_V.

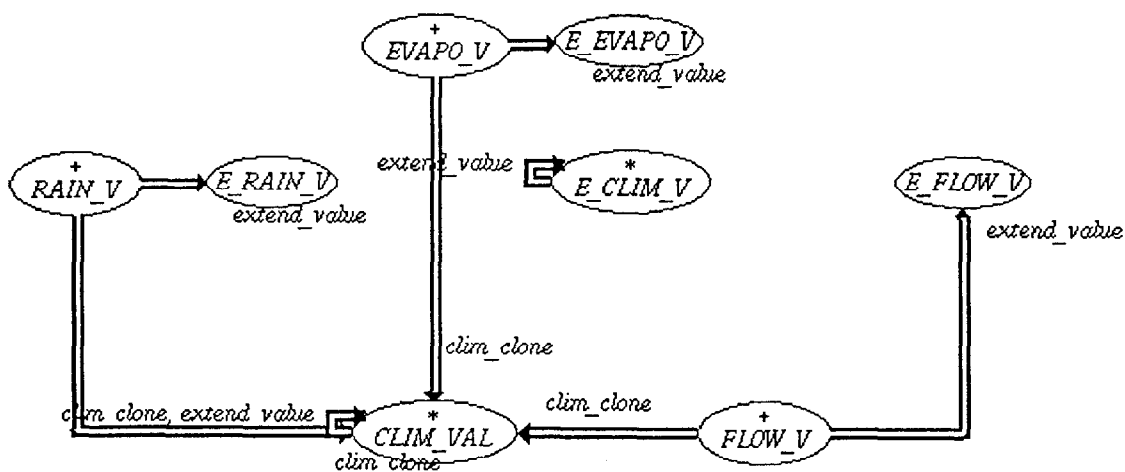
Les classes de ce cluster prennent en compte les valeurs climatiques (pluie, évaporation...).



4.1.18.2. Liens client entre les classes du cluster CLIM_VAI

Les classes RAIN_V, EVAPO_V et FLOW_V sont clientes de la classe CLIM_VAL. La classe FLOW_V est aussi cliente de la classe E_FLOW_V. RAIN_V est cliente de E_RAIN_V.

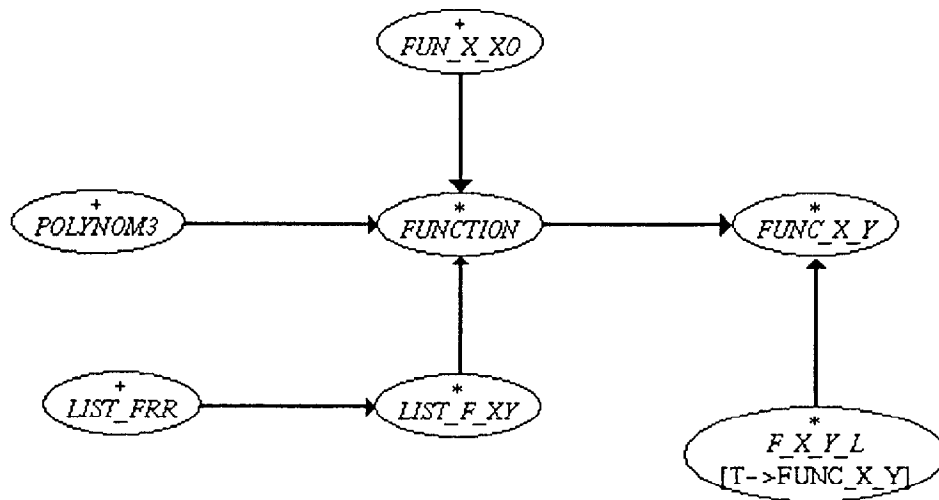
La classe EVAPO_V est cliente de la classe E_EVAPO_V.



4.1.19. Liens d'héritage et client entre les classes du cluster FUNC_X_Y

Dans le cluster FUNC_X_Y, on a les classes FUN_X_XO, POLYNOM3 et LIST_F_XY qui héritent de la classe FUNCTION.

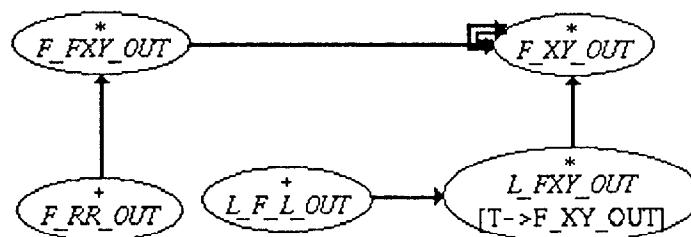
Les classes de ce cluster permettent de prendre en compte les fonctions de la forme $y = f(x)$.



4.1.20. Liens d'héritage et client entre les classes du cluster F_XY_OUT

Dans le cluster F_XY_OUT , les classes $F_FX_Y_OUT$ et $L_FX_Y_OUT[T \rightarrow F_XY_OUT]$ héritent de la classe F_XY_OUT .

Les classes de ce cluster interviennent comme des filtres de sortie de fonctions ou de liste de fonctions.

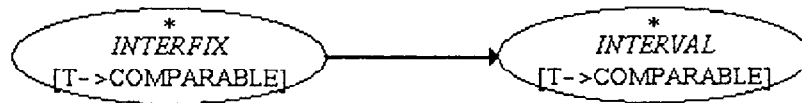


4.1.21. Liens d'héritage et client entre les classes du cluster INTERVAL

Nous avons deux classes dans le cluster INTERVAL: INTERFIX[T→COMPARABLE] et INTERVAL[T→COMPARABLE].

La classe INTERFIX[T→COMPARABLE] hérite de la classe INTERVAL[T→COMPARABLE].

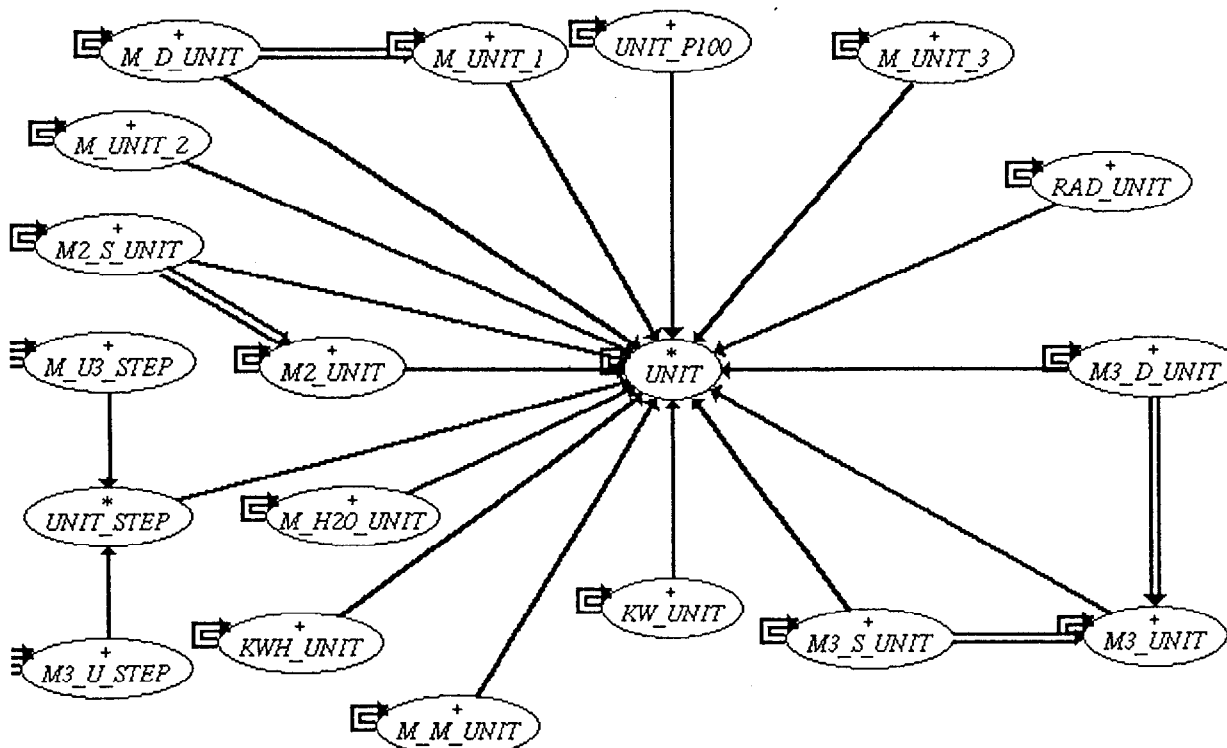
Ces deux classes permettent de définir des intervalles de valeurs numériques.



4.1.22. Liens d'héritage et client entre les classes du cluster UNITS

Dans ce cluster, la classe de base est UNIT. Les autres classes héritent de la classe UNIT.

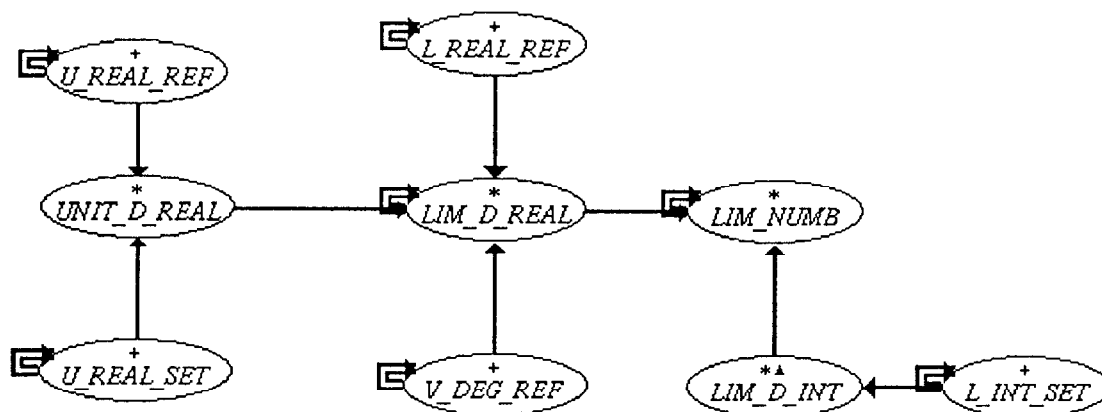
Ce cluster contient toutes les classes définissant et gérant toutes les unités de calculs utilisées dans l'application HYDRAM.



4.1.23. Liens d'héritage et client entre les classes du cluster NUM_VALUE

Les classes UNIT_D_REAL, V_DEG_REF, L_REAL_REF héritent de la classe LIM_D_REAL. Cette même classe hérite de la classe LIM_NUMB.

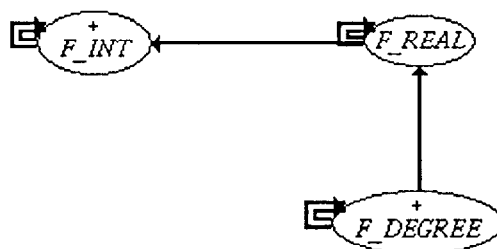
Les classes du cluster NUM_VALUE prennent en compte la valeur et la correspondance avec une chaîne de caractères. Ces classes considèrent des chaînes de caractères (mais qui doivent être des valeurs numériques) saisies par l'utilisateur, les interprètent et trouvent leurs correspondances en valeurs numériques.



4.1.24. Liens d'héritage et client entre les classes du cluster NUM_FORMAT

Dans le cluster NUM_FORMAT, la classe F_DEGREE hérite de la classe F_REAL. La classe F_REAL hérite de F_INT.

Les classes de ce cluster prennent en compte la valeur et la correspondance avec une chaîne de caractères. Ces classes déterminent le format numérique de la chaîne de caractères (format entier, réel ou degré).



4.1.25. Liens d'héritage et client entre les classes du cluster DEG_ANAL

La classe DEG_SCAN hérite de la classe DEG_CONST.

Ces deux classes permettent d'analyser les nombres en degrés.



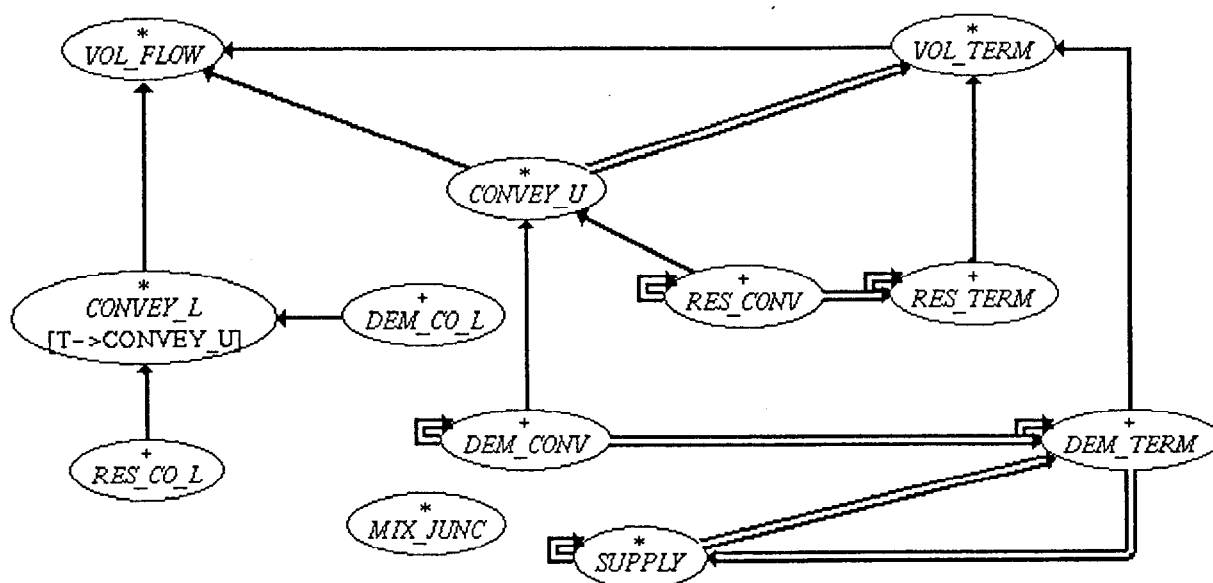
4.1.26. Liens d'héritage et client entre les classes du cluster W_VOL_FLOW

Les classes CONVEY_U, VOL_TERM et CONVEY_L[T→CONVEY_U] héritent de la classe VOL_FLOW.

Les classes RES_TERM et DEM_TERM héritent de la classe VOL_TERM.

La classe CONVEY_U est cliente de la classe VOL_TERM.

Les classes de ce cluster prennent en compte la gestion des volumes et débits dans un système d'eau, le transport d'un débit ou d'un volume d'une ressource à une demande.

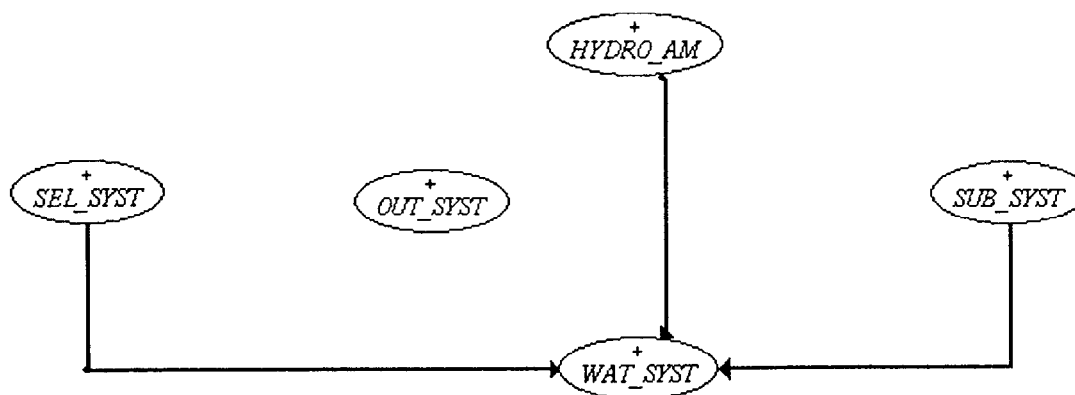


4.1.27. Liens d'héritage et client entre les classes du cluster WAT_SYST

4.1.27.1. Liens d'héritage entre les classes du cluster WAT_SYST

Les classes SEL_SYST, HYDRO_AM et SUB_SYST héritent de la classe WAT_SYST.

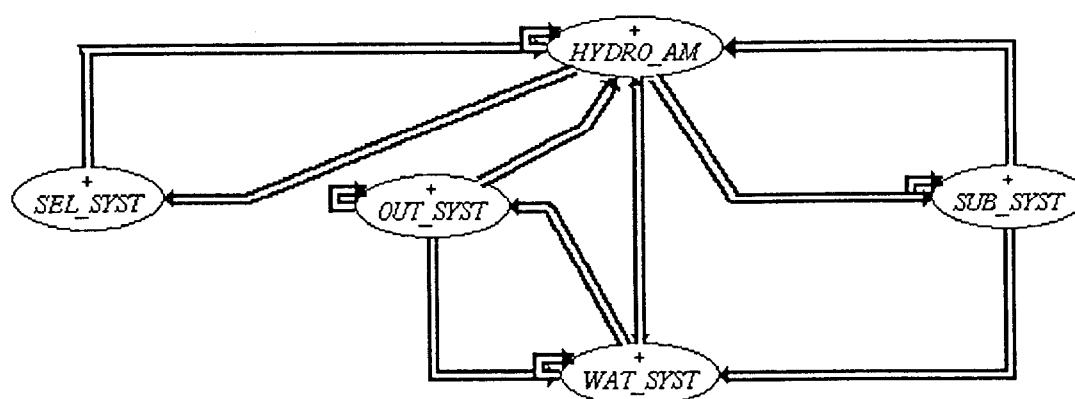
Ces classes de ce cluster prennent en compte la gestion, le stockage des résultats de simulations d'un système d'eau.



4.1.27.2. Liens client entre les classes du cluster WAT_SYST

Les classes SEL_SYST, SUB_SYST, WAT_SYST et OUT_SYST sont clientes de la classe HYDRO_AM.

La classe HYDRO_AM est cliente de la classe SUB_SYST et de la classe SEL_SYST.



4.1.28. Liens d'héritage et client entre les classes du cluster WAT_NODE

4.1.28.1. Liens d'héritage entre les classes du cluster WAT_NODE

Dans le modèle HYDRAM, un système d'eau est composé de noeuds et de liens.

Un noeud d'aménagement peut être une demande, une ressource ou un noeud sans échange avec l'extérieur (par exemple une jonction).

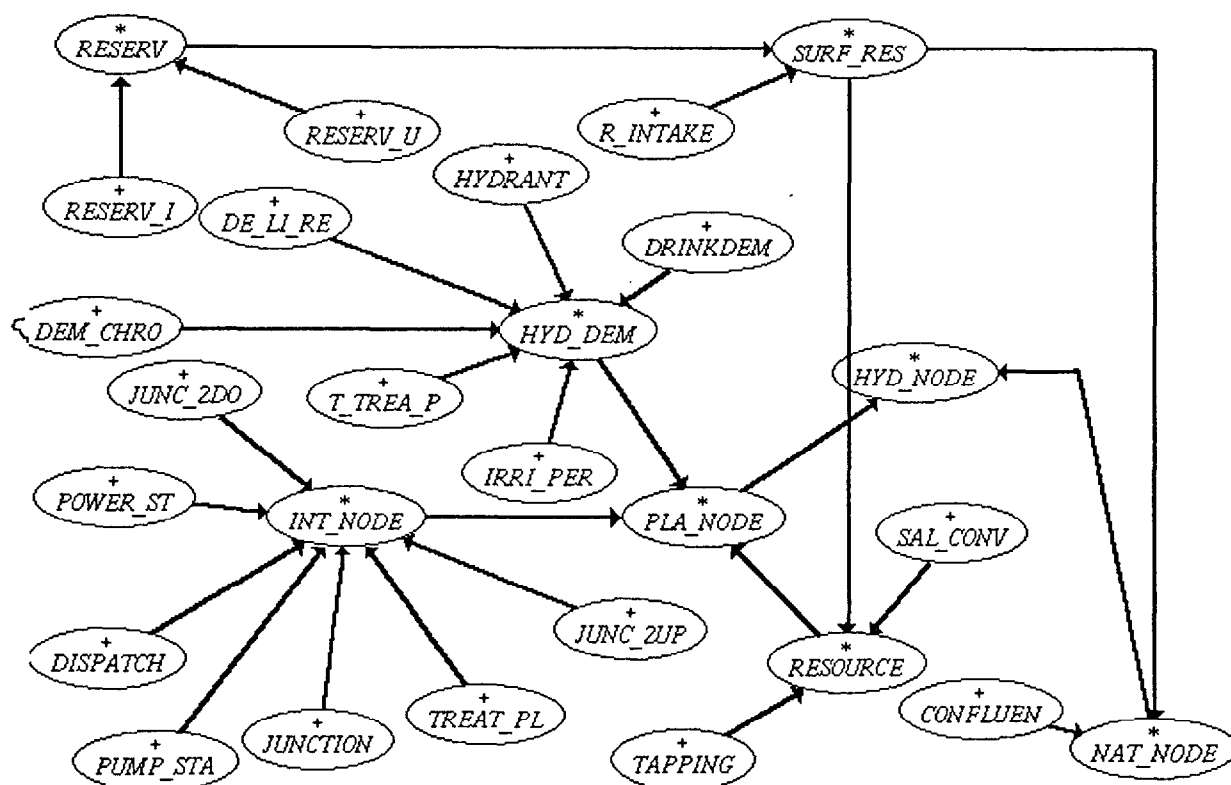
Ainsi les classes HYD_DEM (pour un noeud de type demande), INT_NODE et RESOURCE (pour un noeud de type ressource) héritent de la classe PLA_NODE.

La classe PLA_NODE hérite de la classe HYD_NODE.

Les classes TAPPING, SAL_CONV et SURF_RES héritent de la classe RESOURCE.

Les classes HYDRANT, DE_LI_RE, DEM_CHRO, T_TREA_P, IRRI_PER et DRINKDEM héritent de la classe HYD_DEM.

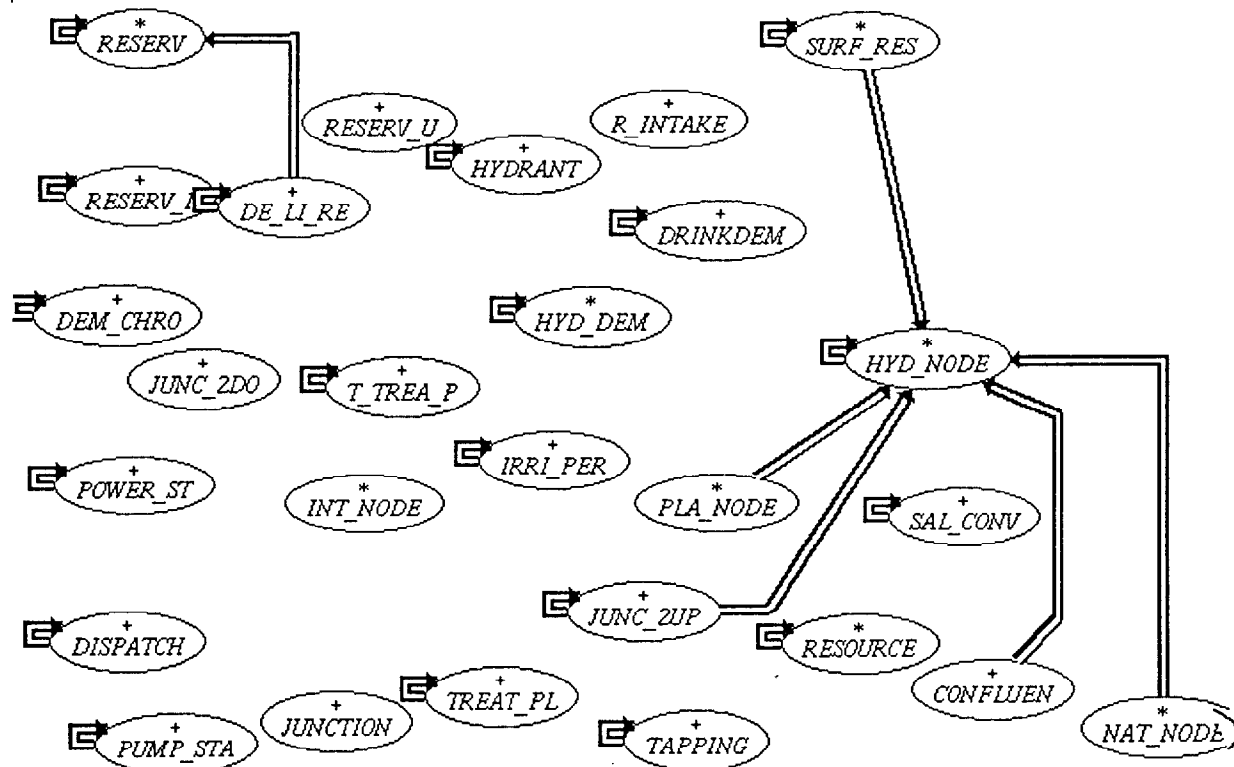
Les classes de ce cluster permettent de définir tous les noeuds disponibles pour l'application HYDRAM.



4.1.28.2. Liens client entre les classes du cluster WAT_NODE

En ce qui concerne les liens client du cluster, les classes SURF_RES, PLA_NODE, JUNC_2UP, CONFLUEN et NAT_NODE sont clientes de la classe HYD_NODE

La classe DE_LI_RE est cliente de la classe RESERV.

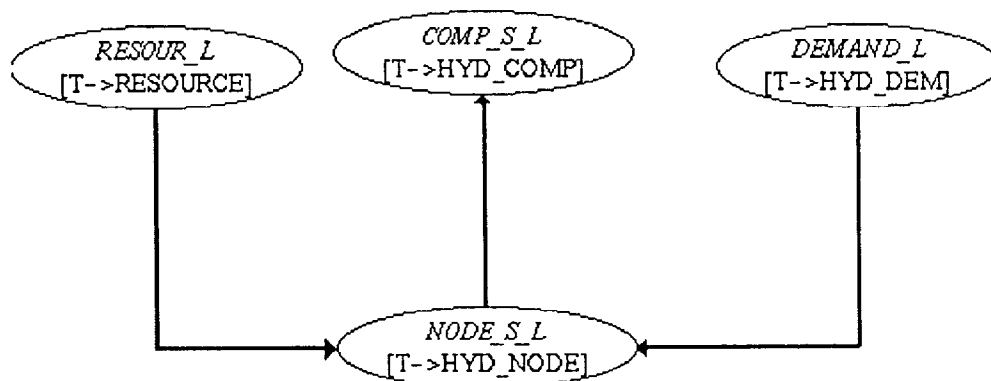


4.1.29. Liens d'héritage et client entre les classes du cluster W_SORT_LIST

Les classes DEMAND_L[T→HYD_DEM] et RESOUR_L[T→RESOURCE] héritent de la classe NODE_S_L[T→HYD_NODE].

La classe NODE_S_L[T→HYD_NODE] hérite de la classe COMP_S_L[T→HYD_COMP].

Les classes du cluster prennent en compte les listes des composants hydrauliques (ressources et demandes).

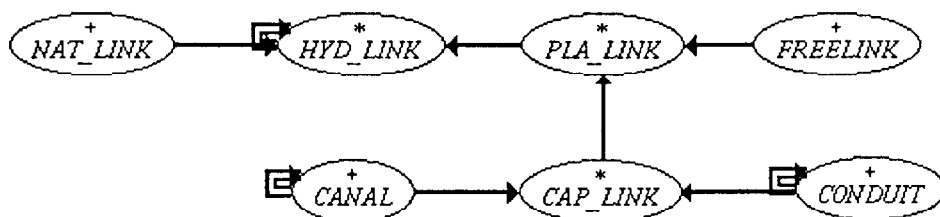


4.1.30. Liens d'héritage et client entre les classes du cluster WAT_LINK

Les classes NAT_LINK et PLA_LINK héritent de la classe HYD_LINK.

Les classes FREELINK et CAP_LINK héritent de la classe PLA_LINK.

Les classes de ce cluster prennent en compte les différents liens d'ouvrages hydrauliques prise en compte dans l'application HYDRAM.



4.1.31. Liens d'héritage et client entre les classes du cluster W_SIM_ENGINE

Les classes IMPROVED_ALLOCATION, SIMPLE_ALLOCATION et HEAD_LOSS_ALLOCATION héritent de la classe SIM_PROC.

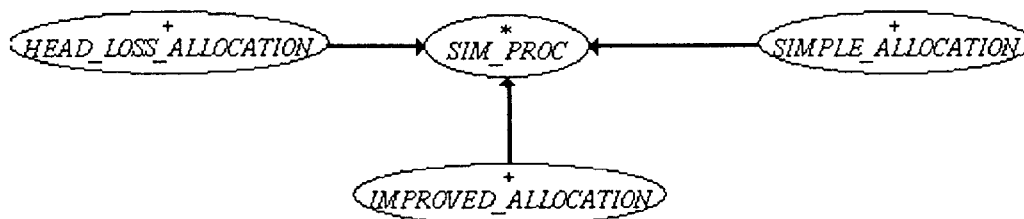
Les classes de ce cluster prennent en compte les différents processus de simulations utilisés dans l'application HYDRAM.

Les différents types de simulations utilisés sont :

la desserte simple

la desserte améliorée

la desserte avec prise en compte des pertes de charges.



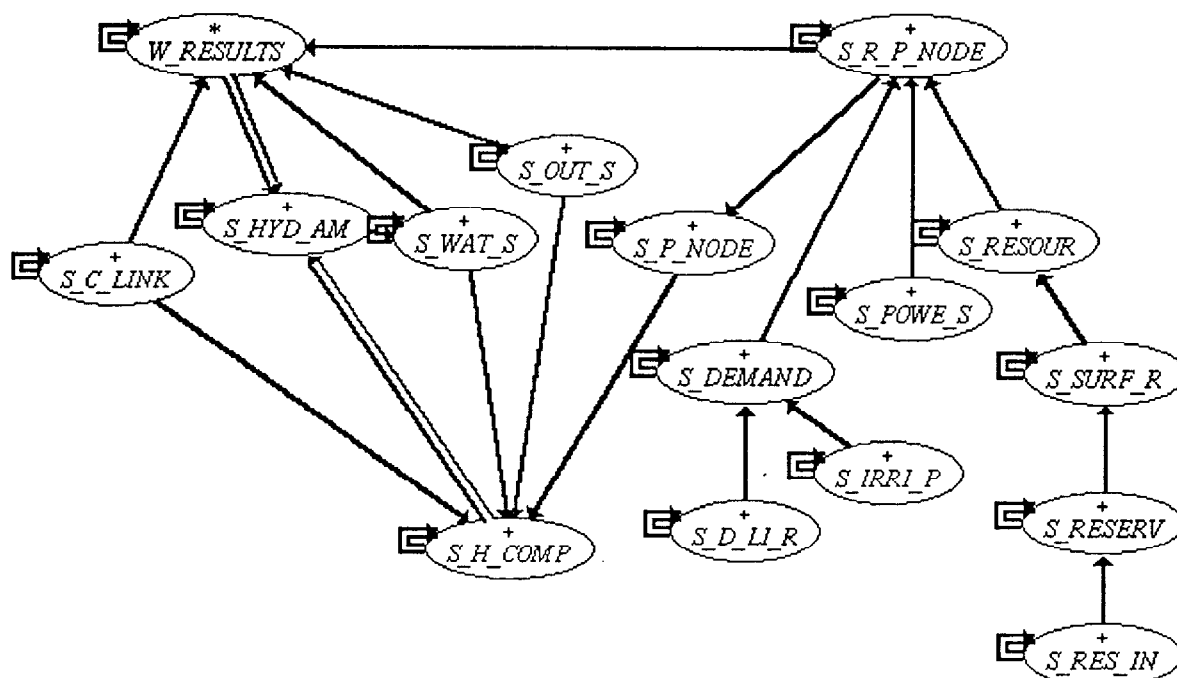
4.1.32. Liens d'héritage et client entre les classes du cluster W_S_COMP

Les classes S_OUT_S, S_WAT_S, S_C_LINK et S_R_P_NODE héritent de la classe W_RESULTS.

Les classes S_RESOUR, S_POWE_S, et S_DEMAND héritent de la classe S_R_P_NODE.

Les classes W_RESULTS et S_H_COMP sont clientes de la classe S_HYD_AM.

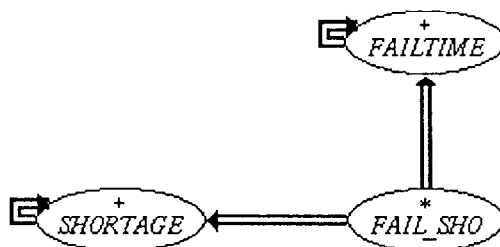
Les classes de ce cluster prennent en compte les données d'un composant hydraulique relatives à une simulation et les résultats relatifs à une simulation d'un composant.



4.1.33. Liens d'héritage et client entre les classes du cluster W_FAIL_SHORT

La classe FAIL_SHO est cliente des classes FAILTIME et SHORTAGE.

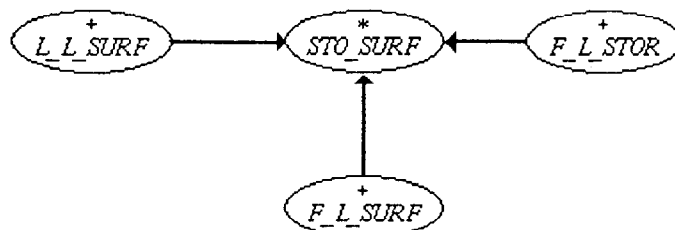
Les classes de ce cluster prennent en compte les pénuries et les durées de défaillances dans la desserte en eau d'un système.



4.1.33. Liens d'héritage et client entre les classes du cluster W_SURF_STO

Les classes L_L_SURF, F_L_STOR et F_L_SURF héritent de la classe STO_SURF.

Les classes de ce cluster prennent en compte le stockage et la surface libre des retenues en fonction du niveau.

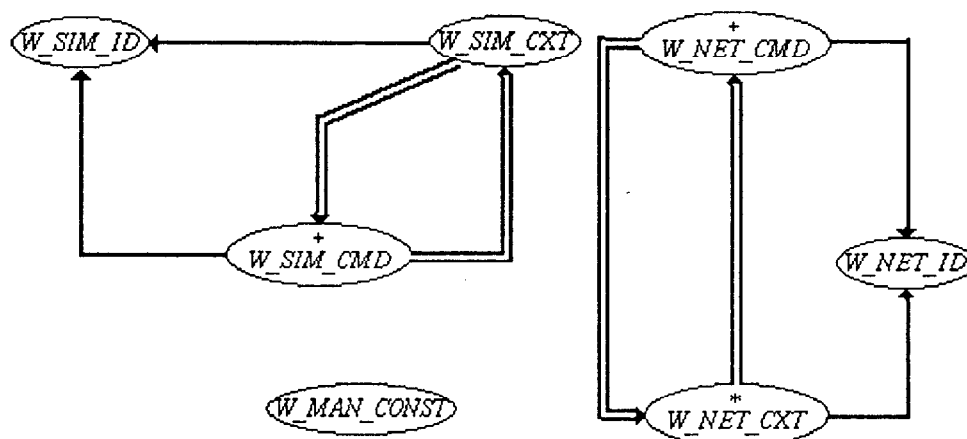


4.1.34. Liens d'héritage et client entre les classes du cluster W_DEF_SET

Les classes W_SIM_CXT et W_SIM_CMD héritent de la classe W_SIM_ID.

Les classes W_NET_CMD et W_NET_CXT héritent de la classe W_NET_ID.

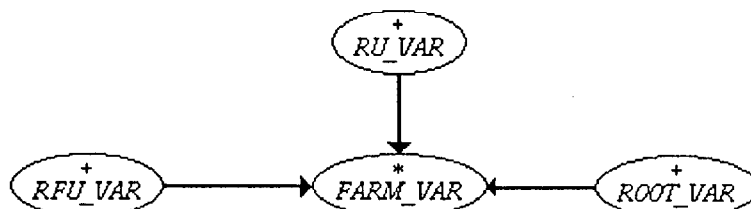
W_SIM_CXT et W_SIM_CMD sont clientes l'une de l'autre et réciproquement.



4.1.35. Liens d'héritage et client entre les classes du cluster W_FARM_VAR

Les classes RU_VAR, RFU_VAR et ROOT_VAR héritent de la classe FARM_VAR.

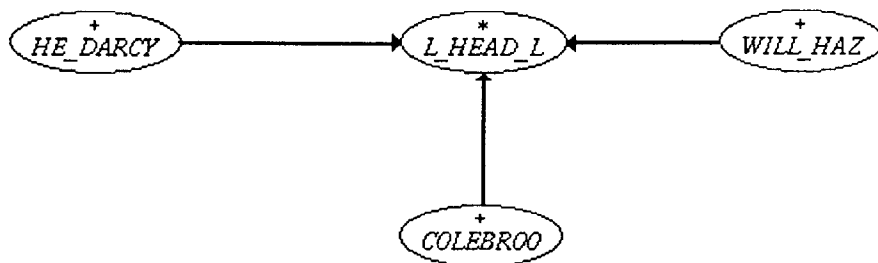
Ces classes prennent en compte les variations de la RU, RFU en fonction du stade cultural et à partir de la variation saisonnière de la profondeur racinaire.



4.1.36. Liens d'héritage et client entre les classes du cluster DISCH_HEAD

Les classes COLEBROO, HE_DARCY et WILL_HAZ héritent de la classe L_HEAD_L.

Ces classes décrivent les formules utilisées dans les calculs des pertes de charges linéaires.



4.2. Liens d'héritage et client entre les classes de l'interface

Les classes de ce cluster décrivent l'interface de l'application HYDRAM.

On y rencontre deux groupes de classes :

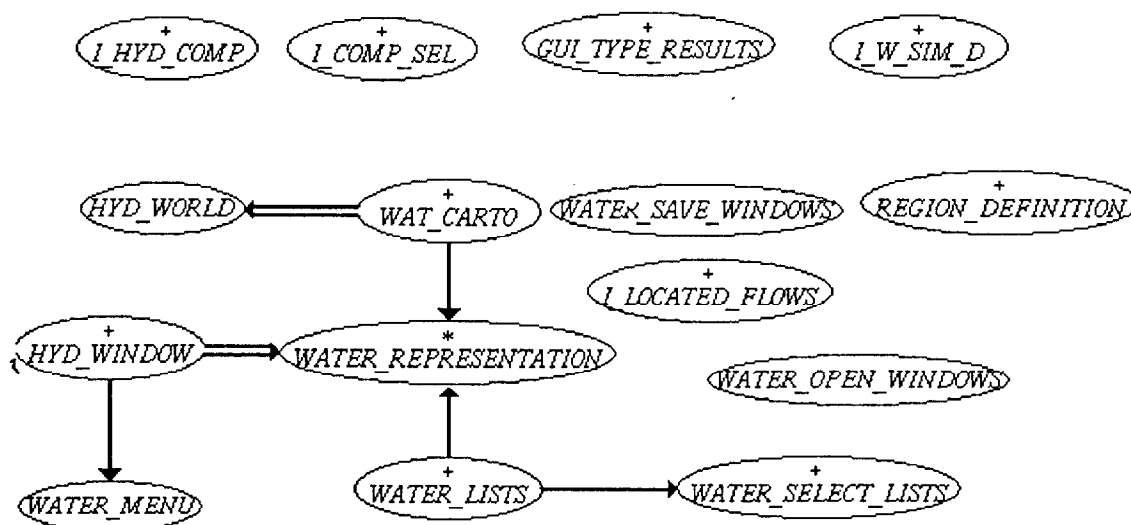
les classes décrivant l'interface elle même, l'effet graphique (c'est à dire tout ce qui peut être contenu dans une fenêtre ; bouton radio, case à cocher, icônes...), ou "motifs".

les classes décrivant des commandes, dont les objets commandes permettent de déclencher des objets cibles.

4.2.1. Liens d'héritage et client entre les classes du cluster I_WAT_MAN_WIDGETS

Dans le cluster I_WAT_MAN_WIDGETS les classes WATER_LISTS et WAT_CARTO héritent de la classe WATER_REPRESENTATION. La classe WATER_LISTS hérite aussi de la classe WATER_SELECT_LISTS.

La classe WAT_CARTO est cliente de la classe HYD_WORLD. HYD_WINDOW est cliente de WATER_REPRESENTATION.

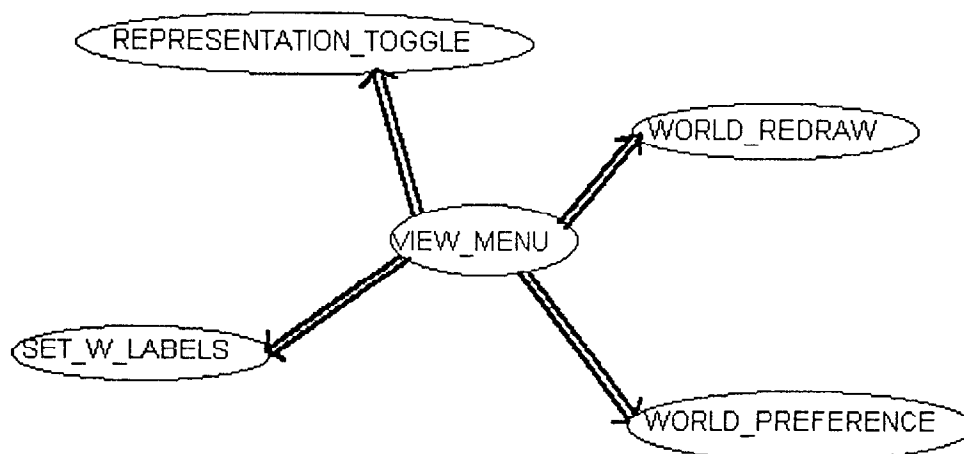


4.2.2. Liens d'héritage et client entre les classes du cluster

I_WAT_MAN_COMMANDS

Nous représentons dans ce cluster, les classes ayant des liens du type héritage ou client avec d'autres classes.

Ainsi la classe VIEW_MENU est cliente des classes REPRESENTATION_TOGGLE, WORLD_REDRAW, SET_W_LABELS et WORLD_PREFERENCE.



4.2.3. Liens d'héritage et client entre les classes du cluster

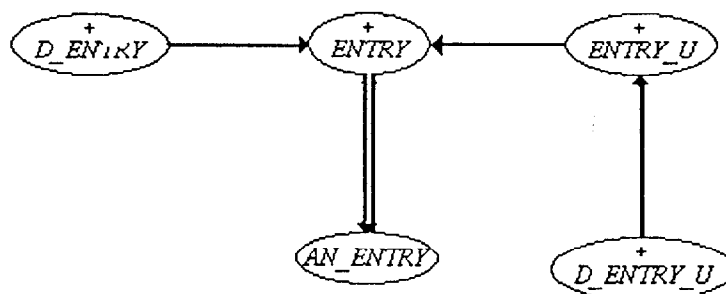
VALUE_FIELD_WIDGETS

Les classes ENTRY_U et D_ENTRY héritent de la classe ENTRY.

La classe D_ENTRY_U hérite de la classe ENTRY_U.

Seule la classe ENTRY est cliente de la classe AN_ENTRY.

Ces classes interviennent au niveau présentation des champs de saisies de l'interface.

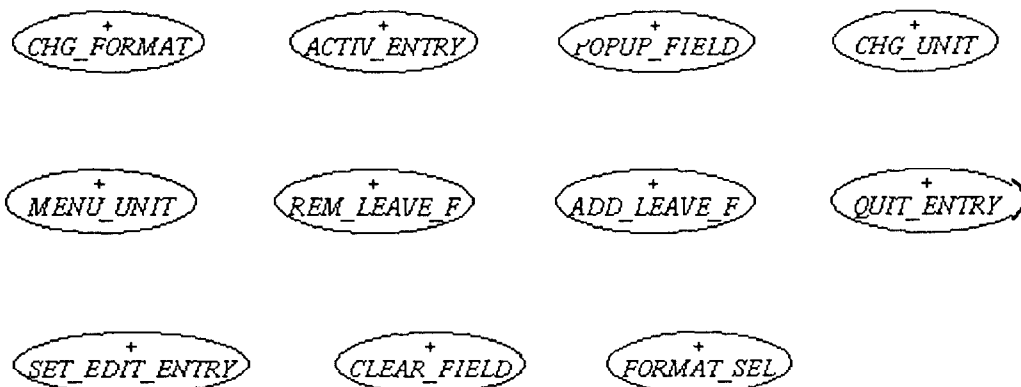


4.2.4. Liens d'héritage et client entre les classes du cluster

VALUE_FIELD_COMMANDS

Il n'existe pas de liens entre les classes de ce cluster VALUE_FIELD_COMMANDS.

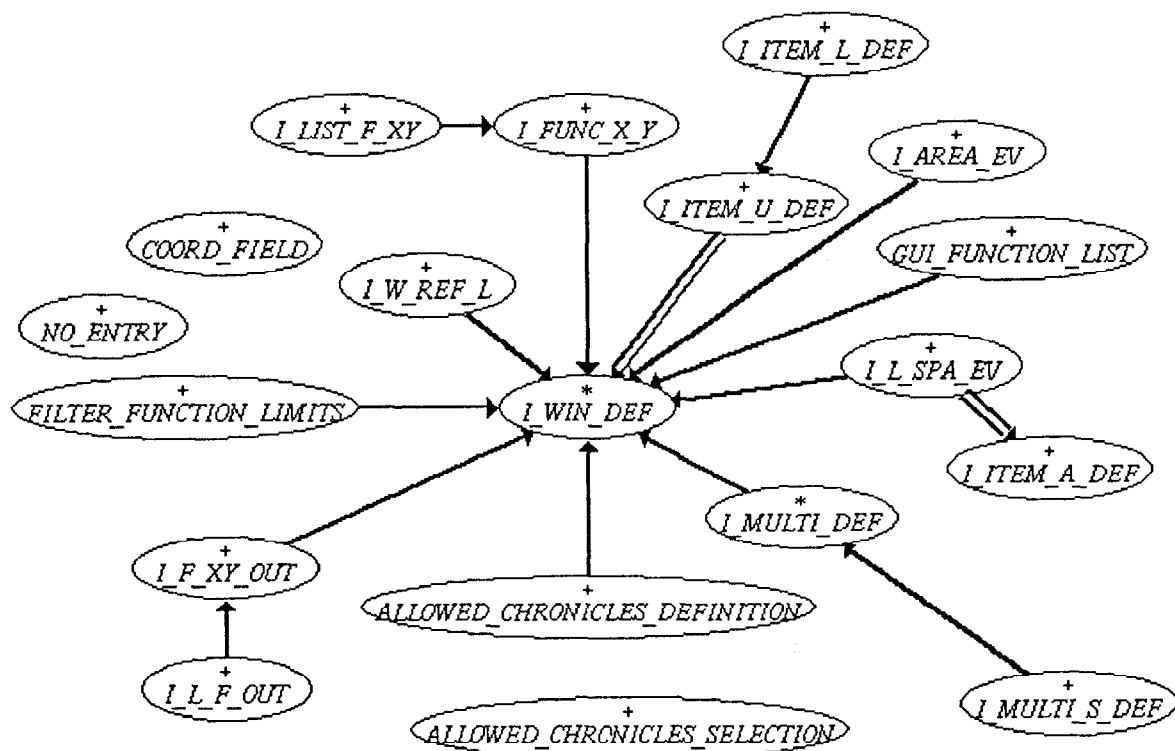
Ces classes interviennent au niveau commande des champs de saisies de l'interface.



4.2.5. Liens d'héritage et client entre les classes du cluster I_OUT_DEF_WIDGETS

La classe de base ce cluster est I_WIN_DEF, une classe différée.

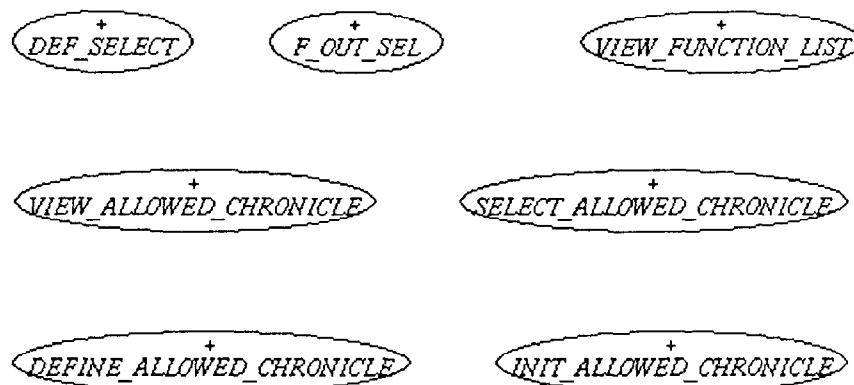
Ces classes permettent de définir des fenêtres de sorties de résultats.



4.2.6. Liens d'héritage et client entre les classes du cluster I_OUT_DEF_COMMANDS

Il n'existe pas de liens entre les classes de ce cluster I_OUT_DEF_COMMANDS.

Ces classes interviennent au niveau commande des fenêtres de sorties des résultats.

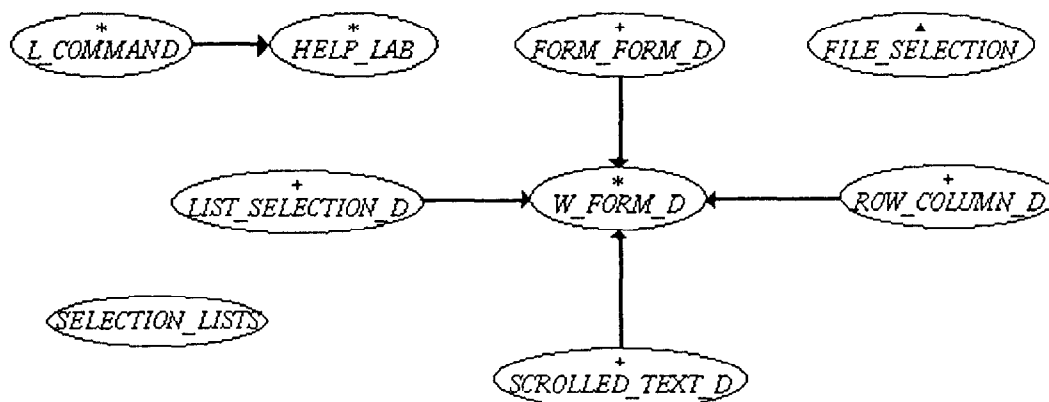


4.2.7. Liens d'héritage et client entre les classes du cluster I_GEN_TOOL_WIDGETS

Les classes LIST_SELECTION_D, FORM_FORM_D, ROW_COLUMN_D et SCROLLED_TEXT_D héritent de la classe W_FORM_D.

La classe L_COMMAND hérite de la classe HELP_LAB.

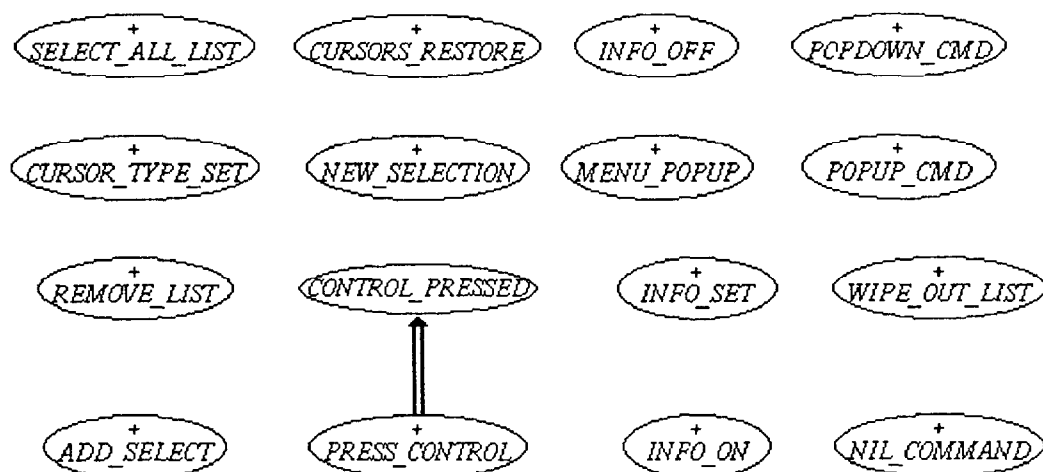
Ces classes forment des outils généraux dans la définition de l'interface de l'application HYDRAM.



4.2.8. Liens d'héritage et client entre les classes du cluster

I_GEN_TOOL_COMMANDS

Il existe un lien de type client de la classe PRESS_CONTROL vers la classe CONTROL_PRESSED. Ces classes forment des outils généraux dans la définition des commandes de l'interface de l'application HYDRAM.



5.1. Structure et composition de l'Ace de l'univers d'HYDRAM

l'Ace de l'univers d'HYDRAM se présente comme suit:

system hydram

root

hydram (interface): "make"

default

assertion (require);
fail_on_rescue (yes);
precompiled ("\$/EIFFEL3/precomp/spec/sparc/mvision");

cluster

-- APPLICATION

appli:	"\$HYD/appli.dir";
wat_man:	"\$HYD/appli.dir/wat_man.dir";
w_out_def:	"\$HYD/appli.dir/wat_man.dir/out_def.dir";
w_def_set:	"\$HYD/appli.dir/wat_man.dir/out_def.dir/def_set.dir";
disch_head:	"\$HYD/appli.dir/wat_man.dir/out_def.dir/dis_head.dir";
w_farm_var:	"\$HYD/appli.dir/wat_man.dir/out_def.dir/farm_var.dir";
w_surf_sto:	"\$HYD/appli.dir/wat_man.dir/out_def.dir/surf_sto.dir";
w_network:	"\$HYD/appli.dir/wat_man.dir/network.dir";
w_sort_list:	"\$HYD/appli.dir/wat_man.dir/network.dir/sortlist.dir";
wat_link:	"\$HYD/appli.dir/wat_man.dir/network.dir/wat_link.dir";
wat_node:	"\$HYD/appli.dir/wat_man.dir/network.dir/wat_node.dir";
wat_syst:	"\$HYD/appli.dir/wat_man.dir/network.dir/wat_syst.dir";
w_sim_parts:	"\$HYD/appli.dir/wat_man.dir/sim_part.dir";
w_fail_short:	"\$HYD/appli.dir/wat_man.dir/sim_part.dir/shortage.dir";
w_request:	"\$HYD/appli.dir/wat_man.dir/sim_part.dir/request.dir";
w_satisf:	"\$HYD/appli.dir/wat_man.dir/sim_part.dir/satisfac.dir";
w_sim_engine:	"\$HYD/appli.dir/wat_man.dir/sim_part.dir/sim_proc.dir";
w_s_comp:	"\$HYD/appli.dir/wat_man.dir/sim_part.dir/sim_comp.dir";
w_vol_flow:	"\$HYD/appli.dir/wat_man.dir/vol_flow.dir";
gen_tool:	"\$HYD/appli.dir/gen_tool.dir";
value_man:	"\$HYD/appli.dir/gen_tool.dir/val_man.dir";
out_def:	"\$HYD/appli.dir/gen_tool.dir/out_def.dir";
value_s:	"\$HYD/appli.dir/gen_tool.dir/val_man.dir/value_s.dir";
format:	"\$HYD/appli.dir/gen_tool.dir/val_man.dir/format.dir";

```

time: "$HYD/appli.dir/time.dir";
time_date: "$HYD/appli.dir/time.dir/date_man.dir";
right_time: "$HYD/appli.dir/time.dir/date_man.dir/date.dir";
timestep: "$HYD/appli.dir/time.dir/date_man.dir/met_step.dir";
date_format: "$HYD/appli.dir/time.dir/date_man.dir/format.dir";
date_anal: "$HYD/appli.dir/time.dir/date_man.dir/analyze.dir";
time_f_man: "$HYD/appli.dir/time.dir/time_fun.dir";
time_f: "$HYD/appli.dir/time.dir/time_fun.dir/function.dir";
time_f_list: "$HYD/appli.dir/time.dir/time_fun.dir/lists.dir";
time_f_out: "$HYD/appli.dir/time.dir/time_fun.dir/func_out.dir";
time_sim: "$HYD/appli.dir/time.dir/time_sim.dir";
time_tool: "$HYD/appli.dir/time.dir/tools.dir";
spatial: "$HYD/appli.dir/spatial.dir";
geo_area: "$HYD/appli.dir/spatial.dir/geo_area.dir";
geo_refer: "$HYD/appli.dir/spatial.dir/geo_ref.dir";
geo_region: "$HYD/appli.dir/spatial.dir/geo_reg.dir";
point: "$HYD/appli.dir/spatial.dir/point.dir";
xyz_tools: "$HYD/appli.dir/spatial.dir/point.dir/xyz_tool.dir";

stations: "$HYD/appli.dir/spatial.dir/stations.dir";
num_tool: "$HYD/appli.dir/num_tool.dir";
f_xy_man: "$HYD/appli.dir/num_tool.dir/f_xy_man.dir";
f_xy_out: "$HYD/appli.dir/num_tool.dir/f_xy_man.dir/f_xy_out.dir";
func_x_y: "$HYD/appli.dir/num_tool.dir/f_xy_man.dir/func_x_y.dir";
n_value_man: "$HYD/appli.dir/num_tool.dir/value_m.dir";
units: "$HYD/appli.dir/num_tool.dir/value_m.dir/units.dir";
deg_anal: "$HYD/appli.dir/num_tool.dir/value_m.dir/deg_anal.dir";
num_format: "$HYD/appli.dir/num_tool.dir/value_m.dir/n_format.dir";
num_value: "$HYD/appli.dir/num_tool.dir/value_m.dir/n_value.dir";
interval: "$HYD/appli.dir/num_tool.dir/interval.dir";
num_list: "$HYD/appli.dir/num_tool.dir/num_list.dir";
spa_time: "$HYD/appli.dir/spa_time.dir";
clim_val: "$HYD/appli.dir/spa_time.dir/clim_val.dir";
spa_ev: "$HYD/appli.dir/spa_time.dir/spa_ev.dir";
spa_ev_man: "$HYD/appli.dir/spa_time.dir/spa_ev_m.dir";
gap_man: "$HYD/appli.dir/spa_time.dir/spa_ev_m.dir/gap_man.dir";
val_man: "$HYD/appli.dir/spa_time.dir/spa_ev_m.dir/val_man.dir";
spa_ev_l: "$HYD/appli.dir/spa_time.dir/spa_ev_l.dir";
spa_ev_f: "$HYD/appli.dir/spa_time.dir/spa_ev_f.dir";
weather: "$HYD/appli.dir/spa_time.dir/weather.dir";

```

-- INTERFACE


```
interface:           "$HYD/inter.dir";
i_wat_man:           "$HYD/inter.dir/wat_man.dir";
i_wat_man_widgets:   "$HYD/inter.dir/wat_man.dir/widgets.dir";
i_wat_man_commands:  "$HYD/inter.dir/wat_man.dir/commands.dir";
i_library:           "$HYD/inter.dir/library.dir";
value_field:         "$HYD/inter.dir/library.dir/value.dir";
value_field_widgets: "$HYD/inter.dir/library.dir/value.dir/widgets.dir";
value_field_commands: "$HYD/inter.dir/library.dir/value.dir/commands.dir";
i_gen_tool:          "$HYD/inter.dir/library.dir/gen_tool.dir";
i_gen_tool_widgets:  "$HYD/inter.dir/library.dir/gen_tool.dir/widgets.dir";
i_gen_tool_commands: "$HYD/inter.dir/library.dir/gen_tool.dir/commands.dir";
i_out_def:           "$HYD/inter.dir/library.dir/out_def.dir";
i_out_def_widgets:   "$HYD/inter.dir/library.dir/out_def.dir/widgets.dir";
i_out_def_commands:  "$HYD/inter.dir/library.dir/out_def.dir/commands.dir";
```

– EiffelBase

```
kernel:              "$EIFFEL3/library/base/kernel";
support:              "$EIFFEL3/library/base/support";
access:               "$EIFFEL3/library/base/structures/access";
cursors:              "$EIFFEL3/library/base/structures/cursors";
cursor_tree:          "$EIFFEL3/library/base/structures/cursor_tree";
dispenser:            "$EIFFEL3/library/base/structures/dispenser";
iteration:             "$EIFFEL3/library/base/structures/iteration";
list:                 "$EIFFEL3/library/base/structures/list";
obsolete:             "$EIFFEL3/library/base/structures/obsolete";
set:                  "$EIFFEL3/library/base/structures/set";
sort:                 "$EIFFEL3/library/base/structures/sort";
storage:              "$EIFFEL3/library/base/structures/storage";
table:                "$EIFFEL3/library/base/structures/table";
traversing:           "$EIFFEL3/library/base/structures/traversing";
tree:                 "$EIFFEL3/library/base/structures/tree";
lexical:              "$EIFFEL3/library/lex";
```

– EiffelVision

```
graph_resources:      "$EIFFEL3/library/vision/implement/motif/Resources";
graph_widgets:        "$EIFFEL3/library/vision/implement/motif/widgets";
graph_toolkit:        "$EIFFEL3/library/vision/implement/toolkit";
graph_X:              "$EIFFEL3/library/vision/implement/X";
graph_commands:       "$EIFFEL3/library/vision/oui/commands";
graph_kernel:         "$EIFFEL3/library/vision/oui/kernel";
graph_oui_widgets:    "$EIFFEL3/library/vision/oui/widgets";
graph_figures:        "$EIFFEL3/library/vision/figures";
```

graph_tools: "\$EIFFEL3/library/vision/tools";

external

 object:

 "\$EIFFEL3/library/vision/spec/\$(PLATFORM)/lib/motif_Clib.a";

 "\$EIFFEL3/library/vision/spec/\$(PLATFORM)/lib/Xt.a";

 "-IXm -IXt -IX11";

 "\$EIFFEL3/library/lex/spec/\$(PLATFORM)/lib/lex.a"

end

5.2. Regroupement des classes par cluster

root: HYDRAM (cluster: interface)	Cluster: w_network (1 class) HYD_COMP	Cluster: wat_syst (5 classes) HYDRO_AM OUT_SYST SEL_SYST SUB_SYST WAT_SYST
Cluster: appli (0 class)	Cluster: w_sort_list (4 classes) COMP_S_L [T -> HYD_COMP] DEMAND_L [T -> HYD_DEM] NODE_S_L [T -> HYD_NODE] RESOUR_L [T -> RESOURCE]	Cluster: w_sim_parts (0 class)
Cluster: wat_man (3 classes) GUAD_SIM HYD_APPL WAT_SIM	Cluster: wat_link (7 classes) CANAL CAP_LINK CONDUIT FREELINK HYD_LINK NAT_LINK PLA_LINK	Cluster: w_fail_short (3 classes) FAIL_SHO FAILTIME SHORTAGE
Cluster: w_out_def (0 class)		Cluster: w_request (1 class) REQUEST
Cluster: w_def_set (7 classes) W_MAN_CONST W_NET_CMD W_NET_CXT W_NET_ID W_SIM_CMD W_SIM_CXT W_SIM_ID	Cluster: wat_node (27 classes) CONFLUEN DE_U_RE DEM_CHRO DISPATCH DRINKDEM HYD_DEM HYD_NODE HYDRANT INT_NODE IRRI_PER JUNC_2DO JUNC_2UP JUNCTION NAT_NODE PLA_NODE POWER_ST PUMP_STA R_INTAKE RESERV RESERV_J RESERV_U RESOURCE SAL_CONV SURF_RES T_TREA_P TAPPING TREAT_PL	Cluster: w_satisf (1 class) SATISFAC
Cluster: disch_head (4 classes) COLEBROO HE_DARCY L_HEAD_L WILL_HAZ		Cluster: w_sim_engine (4 classes) HEAD_LOSS_ALLOCATION IMPROVED_ALLOCATION SIM_PROC SIMPLE_ALLOCATION
Cluster: w_farm_var (4 classes) FARM_VAR RFU_VAR ROOT_VAR RU_VAR		Cluster: w_s_comp (16 classes) S_C_LINK S_D_U_R S_DEMAND S_H_COMP S_HYD_AM S_IRRI_P S_OUT_S S_P_NODE S_POWE_S S_R_P_NODE S_RES_IN S_RESERV S_RESOUR S_SURF_R
Cluster: w_surf_sto (4 classes) F_L_STOR F_L_SURF L_L_SURF STO_SURF		

YEAR_CHRO_L

Cluster: time_f_out (2 classes)

STEP_F_OUT
TIME_F_OUT

Cluster: time_sim (3 classes)

S_TIMETOOL
SIM_TIME
SIMCLOCK

Cluster: time_tool (1 class)

TIMETOOL

Cluster: spatial (0 class)

Cluster: geo_area (3 classes)

GEO_AREA
GEO_CIRC
GEO_RECT

Cluster: geo_refer (1 class)

GEO_REF

Cluster: geo_region (1 class)

REGION

Cluster: point (5 classes)

GEO_XYZ
LON_LAT
X_Y_Z
XY_COORD
XYZ_VECT

Cluster: xyz_tools (5 classes)

G_XYZ_CMD
G_XYZ_CXT
G_XYZ_ID
XYZ_MAN
XYZ_SHAP

Cluster: stations (3 classes)

STA_LIST [T -> STATION]
STATION
STATION_D

Cluster: num_tool (0 class)

Cluster: f_xy_man (0 class)

Cluster: f_xy_out (5 classes)

F_FXY_OUT
F_RR_OUT
F_XY_OUT
L_F_L_OUT
L_FXY_OUT [T -> F_XY_OUT]

Cluster: func_x_y (7 classes)

F_X_Y_L [T -> FUNC_X_Y]
FUN_X_XO
FUNC_X_Y
FUNCTION
LIST_F_XY
LIST_FRR
POLYNOM3

Cluster: n_value_man (0 class)

Cluster: units (23 classes)

D_UNIT
DEG_UNIT
KW_UNIT
KWH_UNIT
M2_S_UNIT
M2_UNIT
M3_D_UNIT
M3_S_UNIT
M3_U_STEP
M3_UNIT
M_D_UNIT
M_H2O_UNIT
M_M_UNIT
M_U3_STEPM_UNIT_1
M_UNIT_2
M_UNIT_3
NO_UNIT
P100_UNIT
RAD_UNIT
UNIT
UNIT_P100
UNIT_STEP

Cluster: deg_anal (2 classes)

DEG_CONST
DEG_SCAN

Cluster: num_format (3 classes)

F_DEGREE
F_INT
F_REAL

Cluster: num_value (12 classes)

INT_1_99
L_INT_SET
L_REAL_REF
L_REAL_SET
LIM_D_INT
LIM_D_REAL
LIM_INT
LIM_NUMB
U_REAL_REF
U_REAL_SET
UNIT_D_REAL
V_DEG_REF

Cluster: interval (2 classes)

INTERFIX [T -> COMPARABLE]
INTERVAL [T -> COMPARABLE]

Cluster: num_list (1 class)

LIST_LI

Cluster: spa_time (0 class)

Cluster: clim_val (8 classes)

CLIM_VAL			
E_CLIM_V			
E_EVAPO_V			
E_FLOW_V			
E_RAIN_V			
EVAPO_V			
FLOW_V			
RAIN_V			
Cluster: spa_ev (7 classes)			
AREA_EV			
E_SPA_EV			
L_SPA_EV			
NIL_EV			
SPA_EV			
SPA_PART_L			
U_SPA_EV			
Cluster: spa_ev_man (0 class)			
Cluster: gap_man (4 classes)			
GAP_MAN			
GAP_MEAN			
GAP_NEAR			
GAP_NO_T			
Cluster: val_man (6 classes)			
V_F_MEAN			
V_MEAN_F			
VAL_CHRO			
VAL_MAN			
VAL_MEAN			
VAL_STEP			
Cluster: spa_ev_l (6 classes)			
GAP_M_CMD			
MEAN_T_CMD			
SPA_CH_L			
SPA_EV_L			
SPA_ME_L			
W_REF_L			
Cluster: spa_ev_f (1 class)			
SPA_EV_F			
Cluster: weather (3 classes)			
	WEATH_AREA		
	WEATHER		
	WEATHER_C		
Cluster: interface (5 classes)			
	CONFIGURATION_FILLER		
	HYDRAM		
	I_CONFIG_P		
	W_WINDOWS		
	WINDOWS		
Cluster: i_wat_man (3 classes)			
	FIGURE_FILLER		
	W_COLOR_CONST		
	W_INTER_ENV		
Cluster: i_wat_man_widgets (16 classes)			
	GUI_TYPE_RESULTS		
	HYD_WINDOW		
	HYD_WORLD		
	I_COMP_SEL		
	I_HYD_COMP		
	I_LOCATED_FLOWS		
	I_W_SM_D		
	ICON_WIN		
	REGION_DEFINITION		
	WAT_CARTO		
	WATER_LISTS		
	WATER_MENU		
	WATER_OPEN_WINDOWS		
	WATER_REPRESENTATION		
	WATER_SAVE_WINDOWS		
	WATER_SELECT_LISTS		
Cluster: i_wat_man_commands (69 classes)			
	BASE_FILE_MENU		
	BASE_VIEW_MENU		
	BUTTON_ZOOM		
	CHRONICLES_SYSTEM		
	CLIP_VIEW		
	CLOSE_HYD		
	COMP_DEF		
	COMPONENT_MENU		
	COMPONENT_VIEW		
	COORD_RESET		
	COORDINATES_SELECT		
	COPY_SEL		
	CUT_SEL		
	DEFINE_LOCATED_FLOWS		
	DEFINE_REGION		
	DEFINE_SELECTION		
	DEL_SEL		
	DISPLAY_PREFERENCES		
	EDIT_MENU		
	EXECUTE_SIMULATION		
	FILE_MENU		
	HELP_MENU		
	HYD_TITLE		
	I_COMPONENT_CHRONICLES		
	I_COMPONENT_RESULTS		
	I_W_SIMUL_UP		
	LOAD_SIMULATION		
	MODIF_MENU		
	NEW_HYD		
	OPEN_HYD		
	PASTE_SEL		
	Q_SAVE_HYD		
	REPRESENTATION_RESET		
	REPRESENTATION_TOGGLE		
	RES_SEL		
	RES_SYST		
	RESIZE_ICONS		
	SAVE_HYD		
	SAVE_HYD_AS		
	SEL_OK		
	SEL_SIM_OK		
	SEL_VIEW		
	SELECT_ALL		
	SELECT_C		
	SELECT_C_L		
	SELECT_DEM		
	SELECT_RES		
	SET_STEP		
	SET_W_LABELS		
	SIMUL_HYD		
	SIMULATION_MENU		
	STORE_OUT		
	SYST_OK		
	SYST_SIM_OK		
	TO_APPL_ITEM		
	TOOLS_MENU		
	VIEW_ALL		
	VIEW_CHRONICLES_SYSTEM		
	VIEW_MENU		
	VIEW_SELECTION		
	W_REF_L_CALCULATE		
	WATER_APPL_SET		
	WATER_ITEM_SELECT		
	WIPE_OUT_SELECTION		
	WORLD_PREFERENCES		
	WORLD_REDRAW		
	ZOOM_MIN		
	ZOOM_PLUS		

ZOOM_VALUE		
Cluster: i_library (0 class)	Cluster: i_gen_tool_commands (20 classes)	DEF_SELECT
	ADD_SELECT	DEFINE_ALLOWED_CHRONICLE
	CONTROL_PRESSED	F_OUT_SEL
	CURSOR_TYPE_SET	INIT_ALLOWED_CHRONICLE
	CURSORS_RESTORE	ITEM_A_DEF_CMD
	DRAW_FIG	ITEM_L_DEF_CMD
Cluster: value_field (0 class)	INFO_OFF	ITEM_U_DEF_CMD
	INFO_OFF_CMD	SELECT_ALLOWED_CHRONICLE
	INFO_ON	VIEW_ALLOWED_CHRONICLE
	INFO_ON_CMD	VIEW_FUNCTION_LIST
Cluster: value_field_widgets (5 classes)	INFO_SET	
AN_ENTRY	MENU_POPUP	Cluster: lexical (24 classes)
D_ENTRY	MESSAGE_INFO	AUTOMATON
D_ENTRY_U	NEW_SELECTION	DFA
ENTRY	NIL_COMMAND	ERROR_LIST
ENTRY_U	POPDOWN_CMD	FIX_AUTOMAT [S -> STATE]
	POPUP_CMD	FIX_INT_SET
	PRESS_CONTROL	FIXED_AUTOMAT [S -> STATE]
	REMOVE_LIST	FIXED_DFA
	SELECT_ALL_LIST	FIXED_INTEGER_SET
	WIPE_OUT_LIST	HIGH_BUILDER
Cluster: value_field_commands (11 classes)		LEX_ARRAY [T]
ACTIV_ENTRY	Cluster: i_out_def (2 classes)	LEX_BUILDER
ADD_LEAVE_F	I_GEN_DEF	LEXICAL
CHG_FORMAT	I_OUT_DEF	LINK_AUTOMAT
CHG_UNIT		LINKED_AUTOMAT [S -> STATE]
CLEAR_FIELD		LINKED_DFA
FORMAT_SEL		METALEX
MENU_UNIT		NDFA
POPUP_FIELD		NFA
QUIT_ENTRY		PDFA
REM_LEAVE_F		SCANNING
SET_EDIT_ENTRY		STATE
		STATE_OF_DFA
		TEXT_FILLER
		TOKEN
Cluster: i_gen_tool (5 classes)	Cluster: i_out_def_widgets (19 classes)	
COLOR_CONST	ALLOWED_CHRONICLES_DEFINITION	Cluster: kernel (Precompiled, 50 classes)
DRAW_WORLD	ALLOWED_CHRONICLES_SELECTION	ANY
FONT_CONST	COORD_FIELD	ARGUMENTS
GEO_WORLD	FILTER_FUNCTION_LIMITS	ARRAY [G]
INTER_ENV	GUI_FUNCTION_LIST	BASIC_ROUTINES
	I_AREA_EV	BIT_REF
	I_F_XY_OUT	BOOLEAN
	I_FUNC_X_Y	BOOLEAN_REF
	I_ITEM_A_DEF	CHARACTER
	I_ITEM_L_DEF	CHARACTER_REF
	I_ITEM_U_DEF	COMPARABLE
	I_L_F_OUT	CONSOLE
	I_L_SPA_EV	DECLARATOR
	I_LIST_F_XY	DIRECTORY
	I_MULTI_DEF	DIRECTORY_NAME
	I_MULTI_S_DEF	DOUBLE
	I_W_REF_L	
	I_WIN_DEF	
	NO_ENTRY	
Cluster: i_gen_tool_widgets (9 classes)	Cluster: i_out_def_commands (10 classes)	
FILE_SELECTION		
FORM_FORM_D		
HELP_LAB		
L_COMMAND		
LIST_SELECTION_D		
ROW_COLUMN_D		
SCROLLED_TEXT_D		
SELECTION_LISTS		
W_FORM_D		

DOUBLE_REF	Cluster: access (Precompiled, 7 classes)	ITERATOR [G]
EXCEP_CONST	ACTIVE [G]	LINEAR_ITERATOR [G]
EXCEPTIONS	BAG [G]	TWO_WAY_CHAIN_ITERATOR [G]
FILE	COLLECTION [G]	
FILE_INFO	CONTAINER [G]	
FILE_NAME	CURSOR_STRUCTURE [G]	
GC_INFO	INDEXABLE [G, H → INTEGER]	
GENERAL	TABLE [G, H]	
HASHABLE		
INTEGER		
INTEGER_REF		
KO_MEDIUM	Cluster: cursors (Precompiled, 9 classes)	Cluster: list (Precompiled, 22 classes)
MEM_CONST	ARRAYED_LIST_CURSOR	ARRAYED_CIRCULAR [G]
MEM_INFO	CIRCULAR_CURSOR	ARRAYED_LIST [G]
MEMORY	COMPACT_TREE_CURSOR	BI_LINKABLE [G]
NUMERIC	CURSOR	CELL [G]
PART_COMPARABLE	LINKED_LIST_CURSOR [G]	CHAIN [G]
PATH_NAME	LINKED_TREE_CURSOR [G]	CIRCULAR [G]
PLAIN_TEXT_FILE	MULTAR_LIST_CURSOR [G]	DYNAMIC_CHAIN [G]
PLATFORM	RECURSIVE_TREE_CURSOR [G]	DYNAMIC_CIRCULAR [G]
POINTER	TWO_WAY_TREE_CURSOR [G]	DYNAMIC_LIST [G]
POINTER_REF		FIXED_LIST [G]
RAW_FILE		LINKABLE [G]
REAL		LINKED_CIRCULAR [G]
REAL_REF		LINKED_LIST [G]
SEQ_STRING	Cluster: cursor_tree (Precompiled, 5 classes)	LIST [G]
SPECIAL [T]	COMPACT_CURSOR_TREE [G]	MULTI_ARRAY_LIST [G]
STD_FILES	CURSOR_TREE [G]	PART_SORTED_LIST [G →
STORABLE	LINKED_CURSOR_TREE [G]	PART_COMPARABLE]
STRING	RECURSIVE_CURSOR_TREE [G]	PART_SORTED_TWO_WAY_LIST [G →
TO_SPECIAL [T]	TWO_WAY_CURSOR_TREE [G]	PART_COMPARABLE]
UNIX_FILE		SEQUENCE [G]
UNIX_FILE_INFO		SORTED_LIST [G → COMPARABLE]
UNIX_SIGNALS		SORTED_TWO_WAY_LIST [G →
UNIX_STD	Cluster: dispenser (Precompiled, 13 classes)	COMPARABLE]
	ARRAYED_QUEUE [G]	TWO_WAY_CIRCULAR [G]
	ARRAYED_STACK [G]	TWO_WAY_LIST [G]
	BOUNDED_QUEUE [G]	
	BOUNDED_STACK [G]	
	DISPENSER [G]	
	HEAP_PRIORITY_QUEUE [G →	
	COMPARABLE]	
	LINKED_PRIORITY_QUEUE [G →	
	COMPARABLE]	
	LINKED_QUEUE [G]	
	LINKED_STACK [G]	
	PRIORITY_QUEUE [G →	
	PART_COMPARABLE]	
	QUEUE [G]	
	SORTED_PRIORITY_QUEUE [G →	
	COMPARABLE]	
	STACK [G]	
	Cluster: iteration (Precompiled, 5 classes)	
	CHAIN_ITERATOR [G]	
	CURSOR_TREE_ITERATOR [G]	
		Cluster: obsolete (Precompiled, 2 classes)
		ARRAY_LIST [G]
		FIXED_CIRCULAR [G]
		Cluster: set (Precompiled, 8 classes)
		BINARY_SEARCH_TREE_SET [G →
		COMPARABLE]
		COMPARABLE_SET [G → COMPARABLE]
		LINKED_SET [G]
		PART_SORTED_SET [G →
		PART_COMPARABLE]
		SET [G]
		SORTED_SET [G → COMPARABLE]
		SUBSET [G]
		TWO_WAY_SORTED_SET [G →
		COMPARABLE]
		Cluster: sort (Precompiled, 2 classes)

Cluster: support (Precompiled, 15 classes)

ASCII
 BOOL_STRING
 COUNTABLE_SEQUENCE [G]
 DOUBLE_MATH
 EXECUTION_ENVIRONMENT
 FIBONACCI
 FORMAT_DOUBLE
 FORMAT_INTEGER
 IDENTIFIED
 INTERNAL
 MATH_CONST
 OPERATING_ENVIRONMENT
 PRIMES
 RANDOM
 SINGLE_MATH

COMPARABLE_STRUCT [G →
COMPARABLE]
SORTED_STRUCT [G → COMPARABLE]

Cluster: storage (Precompiled, 8 classes)

BOUNDED [G]
BOX [G]
COUNTABLE [G]
FINITE [G]
FIXED [G]
INFINITE [G]
RESIZABLE [G]
UNBOUNDED [G]

Cluster: table (Precompiled, 2 classes)

ARRAY2 [G]
HASH_TABLE [G, H → HASHABLE]

Cluster: traversing (Precompiled, 4 classes)

BILINEAR [G]
HIERARCHICAL [G]
LINEAR [G]
TRAVERSABLE [G]

Cluster: tree (Precompiled, 8 classes)

ARRAYED_TREE [G]
BINARY_SEARCH_TREE [G →
COMPARABLE]
BINARY_TREE [G]
DYNAMIC_TREE [G]
FIXED_TREE [G]
LINKED_TREE [G]
TREE [G]
TWO_WAY_TREE [G]

Cluster: graph_resources (Precompiled, 38
classes)

ARROW_B_R_M
BASE_R_M
BULLETIN_R_M
CB_REASONS_R_M
COMPOSITE_R_M
D_AREA_R_M
DIALOG_R_M
DRAW_B_R_M
DRAWN_B_R_M
FILE_SELEC_R_M

FONT_BOX_R_M
FORM_R_M
FRAME_R_M
LABEL_R_M
LIST_R_M
MANAGER_R_M
MENU_B_R_M
MENU_R_M
MESSAGE_R_M
PRIMITIVE_R_M
PROMPT_R_M
PUSH_B_R_M
READ_ACT_R_M
ROW_COLUMN_R_M
SCALE_R_M
SCROLLBAR_R_M
SCROLLED_W_R_M
SEPARATOR_R_M
SHELL_R_M
SIMPLE_M_R_M
TEXT_FIELD_R_M
TEXT_R_M
TOGGLE_R_M
TOP_R_M
TRANSIENT_R_M
WIDGET_R_M
WM_SHELL_R_M
XM_STRING_R_M

Cluster: graph_widgets (Precompiled, 72
classes)

ARROW_B_M
BAR_M
BASE_M
BULLETIN_D_M
BULLETIN_M
BUTTON_M
CHECK_BOX_M
COMPOSITE_M
D_AREA_M
DIALOG_M
DIALOG_S_M
DRAW_B_M
ERROR_D_M
EVENT_HAND_M
FILE_SEL_D_M
FILE_SELEC_M
FONT_B_D_M
FONT_BOX_M
FONTABLE_M
FORM_D_M
FORM_M
FRAME_M
G_ANY_M
INFO_D_M

LABEL_G_M
LABEL_M
LIST_MAN_M
MANAGER_M
MENU_B_M
MENU_M
MENU_PULL_M
MESSAGE_D_M
MESSAGE_M
MOTIF
MOTIF_1
MOTIF_APP
MOUSE_POINTER_M
OPT_PULL_M
OPTION_B_M
OVERRIDE_S_M
PICT_COL_B_M
POPUP_M
POPUP_S_M
PRIMITIVE_M
PROMPT_D_M
PROMPT_M
PULLDOWN_M
PUSH_B_M
PUSH_BG_M
QUESTION_D_M
RADIO_BOX_M
READ_ACTION_M
ROW_COLUMN_M
SCALE_M
SCROLL_L_M
SCROLLBAR_M
SCROLLED_T_M
SCROLLED_W_M
SEPARATO_G_M
SEPARATOR_M
SHELL_M
TERMINAL_M
TEXT_FIELD_M
TEXT_M
TOGGLE_B_M
TOGGLE_BG_M
TOP_M
TOP_SHELL_M
WARNING_D_M
WIDGET_M
WM_SHELL_M
WORKING_D_M

Cluster: graph_toolkit (Precompiled, 79 classes)

ARROW_B_I
BAR_I
BASE_I
BULLETIN_D_I
BULLETIN_I

BUTTON_I
 CHECK_BOX_I
 COLOR_I
 COMPOSITE_I
 D_AREA_I
 DIALOG_I
 DIALOG_S_I
 DRAW_B_I
 DRAWING_I
 ERROR_D_I
 FILE_SEL_D_I
 FILE_SELEC_I
 FONT_B_D_I
 FONT_BOX_I
 FONT_I
 FONT_LIST_I
 FONTABLE_I
 FORM_D_I
 FORM_I
 FRAME_I
 G_ANY_I
 INFO_D_I
 IO_HANDLER_I
 LABEL_G_I
 LABEL_I
 LIST_MAN_I
 MANAGER_I
 MENU_B_I
 MENU_I
 MENU_PULL_I
 MESSAGE_D_I
 MESSAGE_I
 MOUSE_POINTER_I
 OPT_PULL_I
 OPTION_B_I
 OVERRIDE_S_I
 PICT_COL_B_I
 PIXMAP_I
 POPUP_I
 POPUP_S_I
 PRIMITIVE_I
 PROMPT_D_I
 PROMPT_I
 PULLDOWN_I
 PUSH_B_I
 PUSH_BG_I
 QUESTION_D_I
 RADIO_BOX_I
 ROW_COLUMN_I
 SCALE_I
 SCREEN_CURSOR_I
 SCREEN_I
 SCROLL_L_I
 SCROLLBAR_I
 SCROLLED_T_I
 SCROLLED_W_I
 SEPARATO_G_I

SEPARATOR_I
 SHELL_I
 STACKABLE_I
 TASK_I
 TERMINAL_I
 TEXT_FIELD_I
 TEXT_I
 TIMER_I
 TOGGLE_B_I
 TOGGLE_BG_I
 TOP_I
 TOP_SHELL_I
 WARNING_D_I
 WIDGET_I
 WIDGET_RESOURCE_I
 WM_SHELL_I
 WORKING_D_I

Cluster: graph_x (Precompiled, 39 classes)

ALL_CURS_X
 BITMAP_RES_X
 BITMAPS_RES_X
 BUT_CLICK_HAND_X
 BUT_HAND_X
 BUTMT_HAND_X
 COLOR_RES_X
 COLOR_X
 COMPL_HAND_X
 CURSOR_RES_X
 DRAWING_X
 ENTER_HAND_X
 EXP_HAND_X
 FONT_LIST_X
 FONT_RES_X
 FONT_X
 FONTABLE_X
 G_CONTEXT_X
 GRAPHIC_R_X
 IO_HANDLER_X
 KEY_HAND_X
 LEAVE_HAND_X
 MOT_HAND_X
 PIXMAP_RES_X
 PIXMAP_X
 READ_EVENT_R_X
 READ_EVENT_X
 RESOURCE_X
 RESOURCES_X[T]
 SCREEN_CURSOR_X
 SCREEN_X
 SIMPL_HAND_X
 SIZE_HAND_X
 TASK_X
 TIMER_X
 TRANS_HAND_X

TRANS_INFO_X
 WIDGET_RESOURCE_X
 WIDGET_X

Cluster: graph_commands (Precompiled, 55 classes)

BTPRESS_DATA
 BUTCLICK_DATA
 BUTREL_DATA
 BUTTON_DATA
 BUTTONS
 CIRCNOT_DATA
 CIRCREQ_DATA
 CLICK_DATA
 CLRMAP_DATA
 COMMAND
 COMMAND_EXEC
 CONFNOT_DATA
 CONFREQ_DATA
 CONTEXT_DATA
 CREATE_DATA
 DESTROY_DATA
 ENTER_DATA
 EVENT_HDL
 EXPOSE_DATA
 FOCSSOUT_DATA
 FOCUSIN_DATA
 GRAPEXP_DATA
 GRAVNOT_DATA
 HISTORY
 IO_HANDLER
 KEY_DATA
 KEYBOARD
 KEYMAP_DATA
 KEYREL_DATA
 KYPRESS_DATA
 LEAVE_DATA
 MAPNOT_DATA
 MAPPING_DATA
 MAPREQ_DATA
 MESSAGE_DATA
 MODIFY_DATA
 MOTION_DATA
 MOTNOT_DATA
 MOUSE_POINTER
 MULTIPL_DATA
 NOEXP_DATA
 PROPERT_DATA
 REPAREN_DATA
 RESIZE_DATA
 SCALE_DATA
 SELCLR_DATA
 SELNOT_DATA
 SELREQ_DATA
 SINGLE_DATA

TASK
TIMER
TOGGLE_DATA
UNDOABLE
UNMAP_DATA
VISIBLE_DATA

Cluster: graph_kernel (Precompiled, 18 classes)

CLIP
COLOR
COORD_XY
CURSOR_TYPE
DASH
DRAWING
FONT
FONT_LIST
FONTABLE
G_ANY
GRAPHICS
PAINTER
PIXMAP
SCREEN
SCREEN_CURSOR
STACKABLE
TOOLKIT
WIDGET_RESOURCE

Cluster: graph_oui_widgets (Precompiled, 66 classes)

ARROW_B
BAR
BASE
BULLETIN
BULLETIN_D
BUTTON
CHECK_BOX
COMPOSITE
DIALOG
DIALOG_SHELL
DRAW_B
DRAWING_AREA
ERROR_D
FILE_SEL_D
FILE_SELEC
FONT_BOX
FONT_BOX_D
FORM
FORM_D
FRAME
INFO_D
LABEL

LABEL_G
LIST_MAN
MANAGER
MENU
MENU_B
MENU_PULL
MESSAGE
MESSAGE_D
OPT_PULL
OPTION_B
OVERRIDE_S
PCT_COLOR_B
POPUP
POPUP_SHELL
PRIMITIVE
PROMPT
PROMPT_D
PULLDOWN
PUSH_B
PUSH_BG
QUESTION_D
RADIO_BOX
ROW_COLUMN
SCALE
SCROLL_LIST
SCROLLBAR
SCROLLED_T
SCROLLED_W
SEPARATOR
SEPARATOR_G
SHELL
TERMINAL_OUI
TEXT
TEXT_FIELD
TOGGLE_B
TOGGLE_BG
TOP
TOP_SHELL
W_MAN_GEN
W_MANAGER
WARNING_D
WIDGET
WM_SHELL
WORKING_D

Cluster: graph_figures (Precompiled, 45 classes)

ANGLE_ROUT
ARC
ARC_FILLABLE
BACKGROUND
CHILD_CLIP
CIRCLE
CLOSED_FIG

CLOSURE
COMPLEX_FIG
COMPOSITE_FIG
CONFIGURE_NOTIFY
COORD_XY_FIG
DASHABLE
ELLIPSE
ENDED
FIGURE
FILLABLE
FOREGROUND
GEOMETRIC_OP
INF_LINE
INTERIOR
JOINABLE
LINE
LINE_WIDTH
LOGICAL
MOVABLE_FIG
NOTIFY
OPEN_FIG
PATH
PCT_COLOR
PICTURE
POINT
POLYGON
POLYLINE
RECTANGLE
REG_POLYGON
SEGMENT
SLICE
SQUARE
TEXT_FIGURE
TEXT_GEN
TEXT_IMAGE
TRIANGLE
VECTOR
WORLD

Cluster: graph_tools (Precompiled, 12 classes)

DRAW_FIG_COM
HISTORY_CLICK
HISTORY_L_W
HISTORY_LIST
HISTORY_REDO
HISTORY_UNDO
POPDOWN_COM
POPUP_COM
QUIT_APP_D
QUIT_NOW_COM
STD_COMMANDS
UNDOABLE_L

TABLE DES MATIERES

1. INTRODUCTION.....	1
2. DEFINITIONS GENERALES	2
2.1. Classes	2
2.2. Clusters	2
2.3. Lien d'héritage.....	3
2.4. Lien client	3
2.5. Ace	4
3. LIENS D'HERITAGE ET CLIENT ENTRE CLUSTERS	6
3.1. ARBORESCENCE DES CLUSTERS D'HYDRAM	6
3.2. LIENS D'HERITAGE ET CLIENT ENTRE LES CLUSTERS DE L'APPLICATION.....	9
3.2.1. Liens d'héritage et client entre les clusters du cluster APPLI.....	10
3.2.1.1. Liens d'héritage entre les clusters du cluster APPLI.....	10
3.2.1.2. Liens client entre les clusters du cluster APPLI.....	11
3.2.2. Liens d'héritage et client entre les clusters du cluster GEN_TOOL	12
3.2.2.1. Liens d'héritage entre les clusters et classes du cluster GEN_TOOL.....	12
3.2.2.2. Liens client entre les clusters et classes du cluster GEN_TOOL.....	13
3.2.3. Liens d'héritage et client entre les clusters du cluster NUM_TOOL	14
3.2.4. Liens d'héritage et client entre les clusters du cluster WAT_MAN.....	15
3.2.5. Liens d'héritage et client entre les clusters du cluster SPATIAL.....	16
3.2.5.1. Liens d'héritage entre les clusters du cluster SPATIAL.....	16
3.2.5.2. Liens client entre les clusters du cluster SPATIAL	16
3.2.6. Liens d'héritage et client entre les clusters du cluster TIME	17
3.2.6.1. Liens d'héritage entre les clusters du cluster TIME.....	17
3.2.6.2. Liens client entre les clusters du cluster TIME.....	17
3.2.7. Liens d'héritage et client entre les clusters du cluster SPA_TIME.....	18
3.2.7.1. Liens d'héritage entre les clusters du cluster SPA_TIME.....	18
3.2.7.2. Liens client entre les clusters du cluster SPA_TIME	18
3.2.8. Liens d'héritage et client entre les clusters du cluster VALUE_MAN.....	19
3.2.9. Liens d'héritage et client entre les clusters du cluster POINT	19
3.2.10. Liens d'héritage et client entre les clusters du cluster TIME_DATE.....	20
3.2.11. Liens d'héritage et client entre les clusters du cluster TIME_F_MAN	20
3.2.12. Liens d'héritage et client entre les clusters du cluster SPA_EV_MAN	21
3.2.13. Liens d'héritage et client entre les clusters du cluster F_XY_MAN.....	21
3.2.14. Liens d'héritage et client entre les clusters du cluster N_VALUE_MAN.....	22
3.2.15. Liens d'héritage et client entre les clusters du cluster W_NETWORK.....	22
3.2.16. Liens d'héritage et client entre les clusters du cluster W_SIM_PARTS.....	23
3.2.17. Liens d'héritage et client entre les clusters du cluster W_OUT_DEF.....	23
3.3. LIENS D'HERITAGE ET CLIENT ENTRE LES CLUSTERS DE L'INTERFACE.....	24

3.3.1. Liens d'héritage et client entre les clusters du cluster <i>INTERFACE</i>	24
3.3.2. Liens d'héritage et client entre les clusters du cluster <i>I_WAT_MAN</i>	24
3.3.3. Liens d'héritage et client entre les clusters du cluster <i>I_LIBRARY</i>	25
3.3.4. Liens d'héritage et client entre les clusters du cluster <i>VALUE_FIELD</i>	25
3.3.5 Liens d'héritage et client entre les clusters du cluster <i>I_OUT_DEF</i>	25
3.3.6. Liens d'héritage et client entre les clusters du cluster <i>I_GEN_TOOL</i>	26
4. LIENS D'HERITAGE ET CLIENT ENTRE CLASSES	27
4.1. LIENS D'HERITAGE ET CLIENT ENTRE LES CLASSES DE L'APPLICATION	27
4.1.1. Liens d'héritage et client entre les classes du cluster <i>OUT_DEF</i>	27
4.1.2. Liens d'héritage et client entre les classes du cluster <i>VALUE_S</i>	28
4.1.3. Liens d'héritage et client entre les classes du cluster <i>FORMAT</i>	28
4.1.4. Liens d'héritage et client entre les classes du cluster <i>XYZ_TOOLS</i>	29
4.1.5. Liens d'héritage et client entre les classes du cluster <i>STATIONS</i>	29
4.1.6. Liens d'héritage et client entre les classes du cluster <i>TIMESTEP</i>	30
4.1.6.1. Liens d'héritage entre les classes du cluster <i>TIMESTEP</i>	30
4.1.6.2. Liens client entre les classes du cluster <i>TIMESTEP</i>	31
4.1.7. Liens d'héritage et client entre les classes du cluster <i>GEO_AREA</i>	32
4.1.8. Liens d'héritage et client entre les classes du cluster <i>DATE_FORMAT</i>	32
4.1.9. Liens d'héritage et client entre les classes du cluster <i>DATE_ANAL</i>	32
4.1.10. Liens d'héritage et client entre les classes du cluster <i>TIME_F</i>	33
4.1.11. Liens d'héritage et client entre les classes du cluster <i>TIME_F_LIST</i>	33
4.1.11. Liens d'héritage et client entre les classes du cluster <i>TIME_F_OUT</i>	34
4.1.12. Liens d'héritage et client entre les classes du cluster <i>TIME_SIM</i>	34
4.1.13. Liens d'héritage et client entre les classes du cluster <i>WEATHER</i>	34
4.1.14. Liens d'héritage et client entre les classes du cluster <i>SPA_EV</i>	35
4.1.15. Liens d'héritage et client entre les classes du cluster <i>VAL_MAN</i>	35
4.1.16. Liens d'héritage et client entre les classes du cluster <i>GAP_MAN</i>	36
4.1.17. Liens d'héritage et client entre les classes du cluster <i>SPA_EV_L</i>	36
4.1.18. Liens d'héritage et client entre les classes du cluster <i>CLIM_VAI</i>	37
4.1.18.1. Liens d'héritage entre les classes du cluster <i>CLIM_VAI</i>	37
4.1.18.2. Liens client entre les classes du cluster <i>CLIM_VAI</i>	37
4.1.19. Liens d'héritage et client entre les classes du cluster <i>FUNC_X_Y</i>	38
4.1.20. Liens d'héritage et client entre les classes du cluster <i>F_XY_OUT</i>	38
4.1.21. Liens d'héritage et client entre les classes du cluster <i>INTERVAL</i>	39
4.1.22. Liens d'héritage et client entre les classes du cluster <i>UNITS</i>	39
4.1.23. Liens d'héritage et client entre les classes du cluster <i>NUM_VALUE</i>	40
4.1.24. Liens d'héritage et client entre les classes du cluster <i>NUM_FORMAT</i>	40
4.1.25. Liens d'héritage et client entre les classes du cluster <i>DEG_ANAL</i>	41
4.1.26. Liens d'héritage et client entre les classes du cluster <i>W_VOL_FLOW</i>	41
4.1.27. Liens d'héritage et client entre les classes du cluster <i>WAT_SYST</i>	42

4.1.27.1. Liens d'héritage entre les classes du cluster WAT_SYST.....	42
4.1.27.2. Liens client entre les classes du cluster WAT_SYST.....	42
4.1.28. Liens d'héritage et client entre les classes du cluster WAT_NODE.....	43
4.1.28.1. Liens d'héritage entre les classes du cluster WAT_NODE.....	43
4.1.28.2. Liens client entre les classes du cluster WAT_NODE.....	44
4.1.29. Liens d'héritage et client entre les classes du cluster W_SORT_LIST.....	44
4.1.30. Liens d'héritage et client entre les classes du cluster WAT_LINK.....	45
4.1.31. Liens d'héritage et client entre les classes du cluster W_SIM_ENGINE.....	45
4.1.32. Liens d'héritage et client entre les classes du cluster W_S_COMP.....	46
4.1.33. Liens d'héritage et client entre les classes du cluster W_FAIL_SHORT.....	46
4.1.33. Liens d'héritage et client entre les classes du cluster W_SURF_STO.....	47
4.1.34. Liens d'héritage et client entre les classes du cluster W_DEF_SET.....	47
4.1.35. Liens d'héritage et client entre les classes du cluster W_FARM_VAR.....	47
4.1.36. Liens d'héritage et client entre les classes du cluster DISCH_HEAD.....	48
4.2. LIENS D'HERITAGE ET CLIENT ENTRE LES CLASSES DE L'INTERFACE.....	49
4.2.1. Liens d'héritage et client entre les classes du cluster I_WAT_MAN_WIDGETS.....	49
4.2.2. Liens d'héritage et client entre les classes du cluster I_WAT_MAN_COMMANDS.....	50
4.2.3. Liens d'héritage et client entre les classes du cluster VALUE_FIELD_WIDGETS.....	50
4.2.4. Liens d'héritage et client entre les classes du cluster VALUE_FIELD_COMMANDS.....	51
4.2.5. Liens d'héritage et client entre les classes du cluster I_OUT_DEF_WIDGETS.....	51
4.2.6. Liens d'héritage et client entre les classes du cluster I_OUT_DEF_COMMANDS.....	52
4.2.7. Liens d'héritage et client entre les classes du cluster I_GEN_TOOL_WIDGETS.....	52
4.2.8. Liens d'héritage et client entre les classes du cluster I_GEN_TOOL_COMMANDS.....	53
5.1. Structure et composition de l'Ace de l'univers d'HYDRAM.....	54
5.2. Regroupement des classes par cluster.....	58