

# Représentation et Echange de données techniques

**Guy Pierra**

Laboratoire d'Informatique Scientifique et Industrielle

E.N.S.M.A.

86961 Futuroscope Chasseneuil, France

E-mail: pierra@ensma.fr

## Résumé

Avec la généralisation dans le processus de production de la notion de maquette numérique et des outils informatiques qui lui sont associés, se pose le problème de l'archivage à long terme et de l'échange des données techniques qui constitue la nouvelle représentation du savoir-faire de l'entreprise étendue.

L'objet de cette conférence est de présenter un état de l'art sur la modélisation informatique tant des données de produit que des catalogues de composants préexistants utilisés pour concevoir et fabriquer ces produits. On introduit d'abord le langage EXPRESS, développé précisément, pour permettre la représentation des données très structurées propres au domaine technique. Ce langage, développé dans le cadre du projet STEP, est désormais normalisé au niveau ISO. On présente ensuite les modèles développés par le projet STEP pour représenter de façon neutre les données de produit aux différentes phases du cycle de vie et dans les différents secteurs industriels.

On présente ensuite les travaux effectués dans le cadre du projet P-LIB pour permettre la modélisation et l'échange électronique de catalogues de composants industriels pour l'ingénierie et la production entre fournisseurs et utilisateurs. On discute l'état de maturité de ces deux projets, et des normes ISO qui en résultent. On présente enfin les tendances récentes vers l'intégration des approches documentaires (SGML, XML) et base de données (EXPRESS, UML) pour la représentation des bibliothèques et de composants industriels, en particulier destinés à l'Internet, et les synergies existant entre STEP et P-LIB pour une représentation complète des produits.

## Abstract

With the concept of digital mock up, and with all the software tools that generate and use technical data, the extended enterprise needs concepts, methods and tools for exchanging and for long term archiving the digitally-represented knowledge about its products.

The goal of this conference is to present an overview on industrial data modeling and exchanging, both for representing enterprise products and for describing libraries of pre-existing components intended to be used in enterprise products. We first introduce the EXPRESS language, developed within the context of the STEP project (officially ISO 10303), for modeling highly structured industrial data. Then, we outline the information models developed in STEP for the computer-interpretable representation and exchange of product data, throughout the life cycle of a product and for any kind of industry.

In a third section we present the concepts and information models developed within the P-LIB project (officially ISO 13584) for representing and exchanging digital libraries of industrial components. We discuss the achievements of both projects, developed in the same ISO committee. Then we present recent developments performed first, in the context of P-LIB, to integrate the data-oriented view (EXPRESS) and the document-oriented new (HTML/XML) on component libraries and/or electronic catalogues for the web, and, second, to integrate the STEP and P-LIB models and capabilities.

## 1. Introduction

Avec la généralisation, dans le processus de production, c'est-à-dire le processus de conception, fabrication, maintenance et démantèlement des produits industriels, de la représentation informatique des données, et des outils informatiques qui leur sont associés, se pose le problème de l'intégration, de l'échange, et de l'archivage à long terme des données techniques qui constituent le patrimoine informationnel de la nouvelle entreprise.

Concernant les données de produit, deux grandes catégories d'information peuvent être distinguées dans ces données techniques :

- les informations qui décrivent les produits de l'entreprise aux différentes phases de leur cycle de vie et selon les points de vue propres aux différents métiers, et

- les informations qui se réfèrent aux composants préexistants, qui sont utilisés pour concevoir, fabriquer ou maintenir les produits de l'entreprise.

Si ces deux types de données ont de nombreux points communs (par exemple le fait qu'elles décrivent, dans les deux cas, des objets techniques), plusieurs aspects les rendent différents.

- *L'origine* de l'information. L'information sur les produits de l'entreprise est créée dans l'entreprise elle-même. La description et la connaissance sur les composants proviennent de leurs fournisseurs. Permettre un échange purement numérique d'information entre fournisseurs et utilisateurs de composants augmenterait la qualité et réduirait considérablement les coûts de tous les partenaires impliqués dans l'entreprise étendue.
- La *structure* de l'information. Chaque produit réalisé par l'entreprise est individuellement, explicitement et exhaustivement représenté pour permettre sa fabrication. Au contraire, pour faciliter leur recherche et leur sélection, et pour éviter l'explosion des données, les composants sont regroupés en familles d'objets similaires dans lesquelles chaque instance est seulement implicitement décrite à l'aide de tables, d'expressions ou de formules.
- *L'accès* à l'information. Lors d'une recherche d'information sur un produit de l'entreprise, l'utilisateur connaît normalement de quel produit il s'agit. Lorsqu'il cherche un composant pour l'utiliser dans un produit, ce que l'utilisateur connaît est le problème d'ingénierie qu'il doit résoudre. Le système de gestion des données composants doit alors l'aider à trouver le composant adapté pour résoudre son problème. Et, pour ce faire, il doit contenir non seulement les données décrivant les composants, mais également le savoir faire, dont une large partie provient du fournisseur, sur le comportement des composants et sur les critères de sélection.

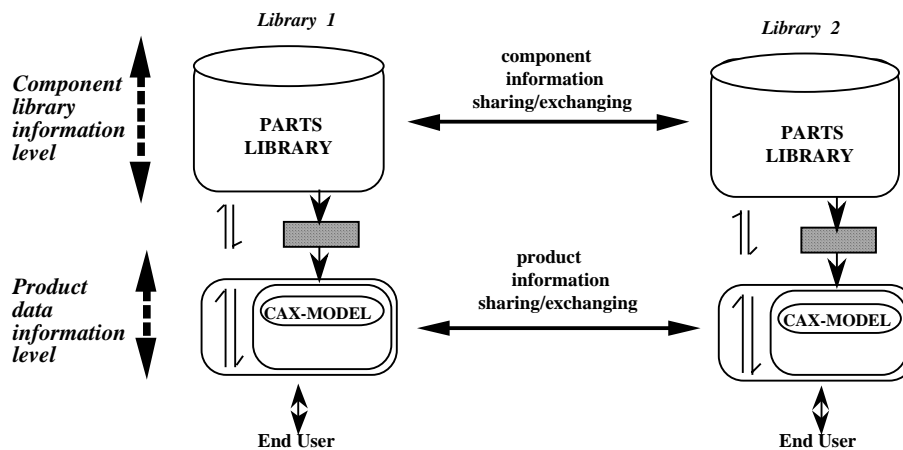


Figure 1 - Les deux niveaux d'échange des données techniques selon [PSP98]

Avec la montée en puissance progressive de la maquette numérique, chaque entreprise se trouve confrontée à la nécessaire définition d'une infrastructure informatique lui permettant de modéliser, de gérer et d'exploiter ces deux types de données techniques. Ceci nécessite, à son tour, de choisir des langages, des modèles, des outils et des systèmes adaptés à ces nouveaux besoins.

L'objet de cette conférence est d'éclairer ce choix en présentant les progrès réalisés au cours des 10 dernières années dans les deux domaines évoqués. Nous nous concentrerons principalement sur les résultats obtenus dans deux grands projets de R & D et de normalisation internationale concernant la modélisation et la gestion des données de produit. Nous présenterons à cette occasion un panorama des langages, outils, modèles et méthodes qui résultent de ces travaux et sont aujourd'hui disponibles pour permettre à chaque entreprise de définir sa propre infrastructure.

Le contenu de cette conférence est le suivant. Dans la première partie, on présente les travaux effectués et les résultats obtenus par le projet international STEP (Standard for the Exchange of Product Data) sur la modélisation des données de produits. Dans la deuxième partie, on présente les travaux effectués sur la modélisation et l'échange de bibliothèques et catalogues de composants. Ce deuxième volet de travaux, initiés en Europe à la fin des années 1980 dans un projet connu sous le nom de CAD-LIB, a ensuite été transféré au niveau international et constitue le projet P-LIB (Parts Library). Enfin une troisième partie présente brièvement les synergies existant entre les deux problématiques et montre l'intérêt d'une gestion intégrée des produits et composants.

## 2. Modélisation de données de produit : la technologie STEP

Dans le processus de conception, l'investissement en matériel et logiciel devient souvent peu de choses par rapport à la valeur que représente les données générées par l'utilisation des systèmes. Le besoin industriel est de disposer des données au bon endroit et au bon moment, alors que tout concourt à l'hétérogénéité des systèmes d'informations et à l'incompatibilité des modèles de données. La diversité des systèmes, qui recouvre la diversité des besoins élémentaires, et leur manque de pérennité,

en particulier dans le contexte de concentration industrielle que nous connaissons, rend à la fois la capitalisation des données difficile (leur durée de vie ne dépasse guère celle du système ayant permis de les générer) et leur échange entre systèmes hétérogènes illusoire (beaucoup du contenu informationnel étant perdu).

C'est dans ce contexte que le projet STEP a été lancé au niveau de l'ISO (International Organization for Standardization) au milieu des années 1980. L'objectif était de définir une représentation non ambiguë des données du produit, interprétable par tout système informatique, et couvrant tout le cycle de vie des produits. Atteindre cet objectif présentait deux difficultés essentielles :

- la définition d'un format neutre, interprétable par tout système informatique indépendamment du système particulier utilisé pour générer les données,
- la couverture d'un très vaste domaine de connaissance, correspondant à l'ensemble des catégories de produits (pièces élémentaires, assemblages, mécanismes, ...), selon le point de vue de tous les métiers (électronique, mécanique, ingénierie, ...), et à toutes les phases du cycle de vie (conception, analyse, fabrication, maintenance, démantèlement). Ceci imposait de mobiliser, et de faire collaborer, des participants représentant un spectre extrêmement large de compétences.

La première difficulté a imposé de développer toute une nouvelle méthodologie sur la modélisation des données pour assurer leur indépendance de tout système. Cette approche comporte :

- la définition d'un langage formel de spécification des données, le langage EXPRESS
- la définition de formats neutres d'échange et de stockage des données décrites dans le langage EXPRESS.

Ces deux éléments font que les données sont destinées à ne plus dépendre des *systèmes* qui les génèrent, mais des *modèles* qui les décrivent.

La deuxième difficulté a amené à élaborer des procédures permettant de faire collaborer des spécialistes de chacun des domaines techniques - caractérisés par les produits concernés, la phase du cycle de vie prise en compte, et le point de vue disciplinaire adopté- avec des spécialistes de la modélisation des données pour construire les modèles propres à chaque domaine. Elle a ensuite amené au développement d'une méthodologie spécifique de modélisation visant à permettre à ces nombreux groupes d'experts de mettre en commun les modèles relevant des aspects communs.

Après plus de 15 ans de travaux, au cours desquels plusieurs centaines d'experts appartenant à 19 pays (Australie, Belgique, Brésil, Canada, Chine, France, Allemagne, Hongrie, Italie, Japon, Corée du Sud, Pays-Bas, Norvège, Roumaine, Russie, Suède, Suisse, Royaume-Uni, Etats-Unis) se sont réunis plus de trois fois par an, une véritable technologie a été développée. Cette norme est décrite dans les différents fascicules de la norme STEP, officiellement la norme ISO 10303. Celle-ci qui comporte actuellement, publié ou encore en travaux, environ 110 fascicules soit quelques dizaines de milliers de pages de spécifications.

## 2.1 Les différents aspects de la technologie STEP

La figure ci-dessous présente les différentes parties de la technologie développée, qui correspondent également à la classification des différents fascicules publiés ou encore en travaux.

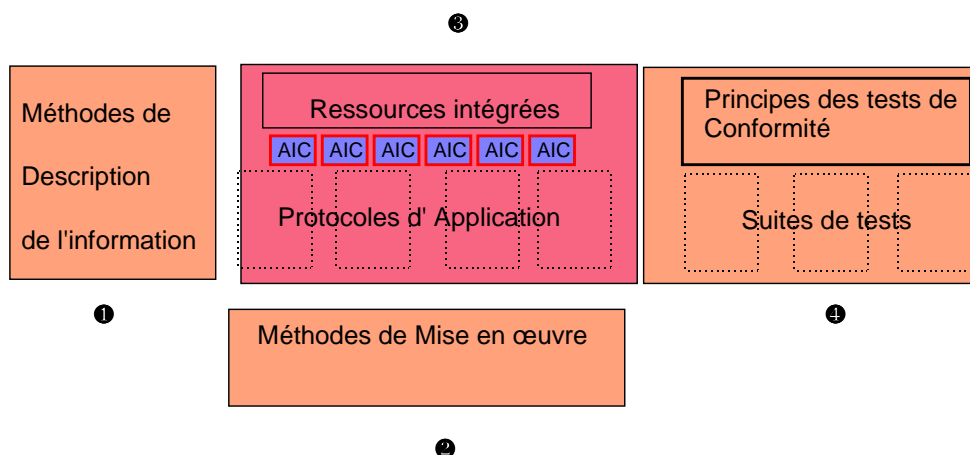


Figure 2 – Structure de la norme STEP

Nous détaillons ci-après le contenu de chacune de ces parties (on pourra visiter le site <http://www.nist.gov/sc4/> qui est le site de référence pour STEP).

## 2.2 Méthodes de description de l'information: le langage EXPRESS

Le travail sur le développement des modèles STEP a commencé en 1986. Aucune notation de modélisation des données ne s'avérant à la fois suffisamment précise et suffisamment expressive [SW94] un nouveau langage dû être développé.

## 2.2.1 Origine

EXPRESS [ISO10303.11-94] est le langage de modélisation des données conçu dans le cadre du projet STEP. Son objectif principal est la description de modèles d'informations dans le domaine technique, en vue de l'échange de données représentant de façon fiable et non ambiguë ces informations [SW94] [BOU95]. Dans le langage EXPRESS, l'accent principal est mis sur la précision du modèle et tout particulièrement sur les contraintes que doivent respecter les données pour être acceptées comme conforme au modèle, ceci pour assurer la fiabilité de l'information représentée. EXPRESS n'est pas seulement une notation permettant la *modélisation* des données, c'est à dire une représentation simplifiée, éventuellement partiellement ambiguë, des informations propres à un domaine a fins d'échange entre concepteurs humains pour décider des éléments qui sont pertinents et des détails qu'il convient de négliger. C'est également un formalisme de *spécification*, c'est à dire qu'il permet une description complètement non ambiguë et traitable par machine.

En fait (ainsi que le montre l'exemple que nous traiterons ci-dessous), le langage EXPRESS peut être utilisé pour la spécification des données dans de très nombreux domaines autres que la modélisation de produits. Par contre, à la différence des formalismes de modélisation objets, tel OMT ou UML [RBP91] [JBR99] il n'est pas destiné à modéliser des systèmes informatiques faisant intervenir une composante dynamique [MEY88]. En EXPRESS, les objets ne possèdent pas de méthodes, et la connaissance procédurale représentable s'exprime exclusivement soit sous forme de contraintes d'intégrité, soit sous forme de fonctions de dérivation exprimant comment un attribut se calcule à partir d'autres attributs.

## 2.2.2 Le langage EXPRESS

Un modèle EXPRESS définit un ensemble d'entités qui représentent les objets à modéliser. EXPRESS suit une approche orientée type [ATK 90] : les types d'entités sont définis dans un schéma statique, et il n'y a pas de notion de métaclasse. On peut cependant, en travaillant au niveau méta [ABG95] [APS95], modéliser le schéma lui-même (c'est la méthode développée dans le projet P-LIB pour représenter le comportement des composants d'un catalogue).

Chaque entité est définie par un ensemble de caractéristiques appelées attributs. Chaque attribut possède un domaine de valeurs licites (type de données, pouvant eux-mêmes être des entités). EXPRESS permet de préciser ces domaines grâce à des contraintes d'intégrité. Enfin, conformément à l'approche objet, [MEY88] [CY92] les entités sont organisées de façon hiérarchique par des relations de généralisation/spécialisation associées à un mécanisme de factorisation/héritage.

L'exemple de la figure suivante présente une arborescence d'entités : deux sous-types d'entités, *étudiant* et *enseignant*, sont des spécialisations d'un super-type d'entités non instanciable (mot clé ABSTRACT), *personne*.

```
ENTITY personne
ABSTRACT SUPERTYPE OF
  ONEOF (étudiant, enseignant);
END_ENTITY;

ENTITY étudiant
SUBTYPE OF personne;
END_ENTITY;

ENTITY enseignant
SUBTYPE OF personne;
END_ENTITY;
```

Figure 3 – Un exemple de hiérarchie de classes EXPRESS

Le langage EXPRESS permet de distinguer les instances d'entités et les simples valeurs. Ces dernières peuvent appartenir aux types prédéfinis usuels (entier, réel, chaîne,...). Il est également possible de construire des nouveaux types dit "types utilisateurs" (mot clé **TYPE**). Ceci permet de donner un nom différent à un type existant, ou de construire, soit des types énumérés et soit des types sélections (qui définissent une union de types).

La principale originalité d'EXPRESS est au niveau de la représentation des contraintes. EXPRESS dispose seulement de deux abstractions de haut niveau qui expriment les *invariants des données*, respectivement sous forme fonctionnelle et assertionnelle. La *dérivation* permet de calculer de façon fonctionnelle un attribut à partir des autres éléments du modèle. La *contrainte d'intégrité*, locale (clause WHERE) ou globale (GLOBAL), s'exprime sous forme de prédicat qui doit être vrai pour toute donnée licite. Un langage de programmation, de type impératif et similaire à PASCAL, permet d'exprimer ces invariants à l'aide de toute fonction programmable.

Du point de vue syntaxique, l'ensemble des constructions qui visent à représenter un univers est regroupé dans un SCHEMA. Du point de vue sémantique, ce schéma définit avec beaucoup de précision un ensemble de valeurs licites qui en constitue le domaine possible d'interprétation (i.e., l'ensemble de valeurs autorisées).

Après adjonction de types de données, d'attributs et des contraintes nécessaires pour préciser les propriétés que doivent respecter toutes les populations de données conformes au modèle, le schéma précédent, deviendrait par exemple :

```

SCHEMA etablissement_schema;

TYPE t_salaire = REAL;
WHERE
  SELF > 0;
END_TYPE;

TYPE t_num_ss = STRING(13) FIXED;
WHERE
  wr: SELF[1] = '1'
  OR SELF[1] = '2';
END_TYPE;

TYPE t_cours = ENUMERATION OF(
  math, info, histoire, sport);
END_TYPE;

TYPE t_nom = STRING;
WHERE
  LENGHT(SELF) > 0;
END_TYPE;

ENTITY personne
ABSTRACT SUPERTYPE OF
  ONEOF (etudiant, enseignant);
  num_ss: t_num_ss;
  nom: t_nom;
DERIVE
  initiale:STRING :=
    calcule(SELF.nom);
UNIQUE
  ur: num_ss;
END_ENTITY;

ENTITY etudiant
SUBTYPE OF personne;
  notes: ARRAY [1:5] OF OPTIONAL REAL;
  suit: ARRAY [1:5] OF OPTIONAL cours;
WHERE
  wr: SIZEOF(QUERY(x <* notes |
    (x < 0) OR (x > 20))) = 0;
END_ENTITY;

ENTITY enseignant
SUBTYPE OF personne;
  salaire: OPTIONAL t_salaire;
  enseigne: SET[1:?] OF cours;
END_ENTITY;

ENTITY cours
  intitulé: t_cours;
INVERSE
  est_suivi_par: SET[0:30] OF
    etudiant FOR suit;
  est_enseigné_par: enseignant
    FOR enseigne;
END_ENTITY;

FUNCTION calcule(nom: t_nom): STRING;
  RETURN(nom[1]);
END_FUNCTION;

END_SCHEMA ; -- etablissement_schema

```

Figure 4 – Un modèle de données contraint

Toute attribut, si cela est permis par le modèle (mot clé : OPTIONAL) y compris dans un tableau (mots clés : ARRAY OF OPTIONAL), peut être autorisée à avoir la valeur "indeterminate". La clause WHERE introduit une contrainte d'intégrité locale destinée à être vérifiée pour chaque instance d'un type ou d'une entité (les numéros de sécurité sociale ne commençant pas par '1' ou '2' sont interdits). La clause UNIQUE, contrainte d'intégrité globale au modèle mais exprimable localement, indique l'unicité des valeurs d'un ou plusieurs attributs pour toutes les instances d'entités d'un type d'entités (ici, il ne peut pas exister plusieurs personnes ayant le même numéro de sécurité sociale). La clause INVERSE permet d'associer à toute relation, la relation inverse. Du point de vue sémantique, il s'agit d'une méthode de dérivation implicite qui évalue, pour chaque instance, l'ensemble des instances qui y font référence dans un rôle donné. Ce lien inverse possède deux utilisations. D'une part, il permet d'exprimer la cardinalité d'une relation inverse (tout cours est enseigné par un enseignant et un seul ; il est suivi par un maximum de 30 élèves ; dans un modèle géométrique de type B-rep, toute arête appartient à exactement deux faces). D'autre part, il permet d'accéder à partir d'une instance, par la notation pointée, à toutes les instances qui y font référence (par exemple pour savoir quels étudiants suivent un cours donné, ou, dans un modèle géométrique de type B-rep, à quelle face appartient une arête particulière). Enfin, des fonctions prédéfinies permettent d'accéder à l'ensemble des instances d'un type d'entité ou d'un agrégat (ensembles, listes,...), de les parcourir de façon indicée (LOBOUND...HIBOUND) ou globale (clause QUERY), de calculer la taille d'un ensemble (clause SIZEOF), d'accéder au type d'une entité, et, pratiquement, d'exprimer toute contrainte susceptible de s'écrire sous forme d'un programme de type PASCAL (voir par exemple la fonction *calcule* qui évalue l'initiale d'un nom en en prenant le premier caractère).

### 2.2.3 La notation graphique EXPRESS-G

La représentation textuelle des schémas EXPRESS est essentielle pour le traitement automatique des modèles. C'est elle qui pourra être exploitée par machine pour vérifier sa correction, ou vérifier si des instances lui sont conformes (cf. 2.3). Elle est, par contre, assez difficilement lisible. Un formalisme graphique, EXPRESS-G [ISO10303.11-94] [SW94], a donc été défini pour donner une vue synthétique des modèles de données et faciliter leur conception dans les phases initiales d'analyse des problèmes à modéliser. Ce symbolisme de représentation n'est que partiel et ne permet pas d'exprimer les contraintes d'intégrité. La nature des attributs (optionnel, dérivé ou inverse), ainsi que leur type, peuvent, par contre, être représentés. La représentation en EXPRESS-G du modèle de données précédent est donnée ci-dessous :

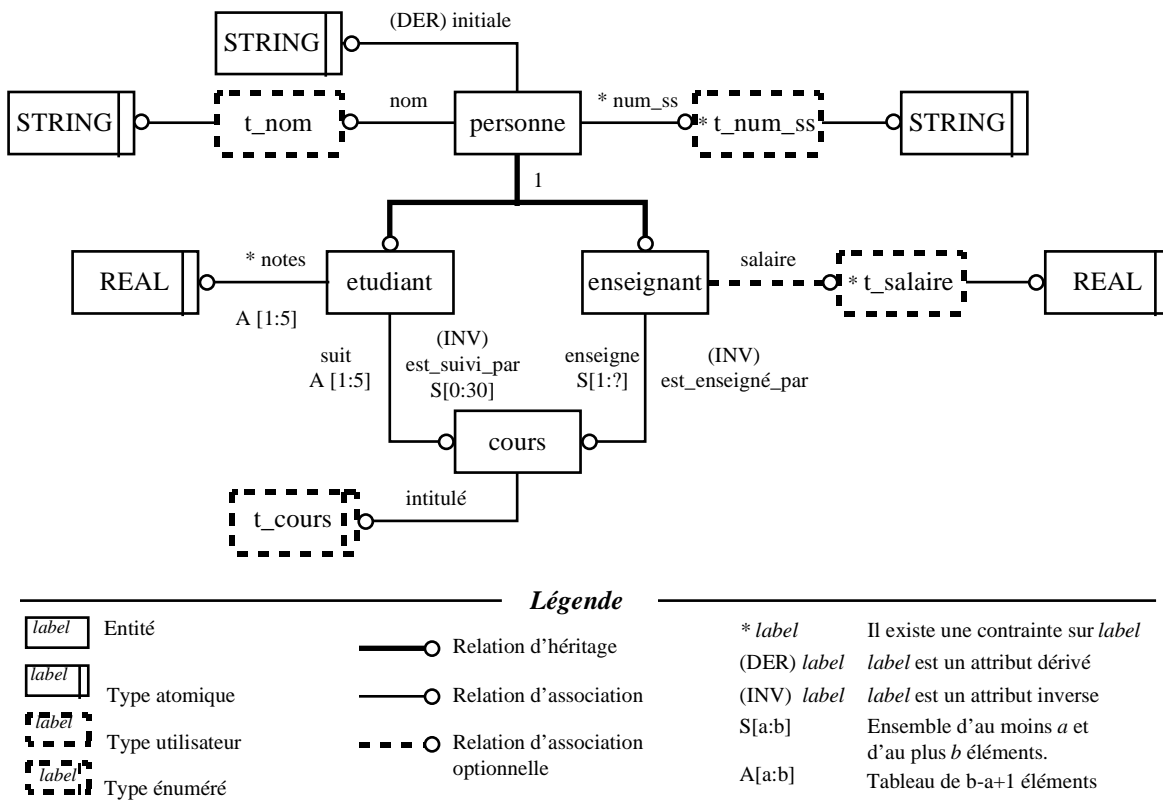


Figure 5 – Représentation en EXPRESS-G d'un modèle de données [SAR99]

## 2.2.4 Intérêt d'EXPRESS

A la différence des nombreuses notations existantes pour modéliser des données telles que NIAM [HAB93] OMT [RBP91] ou UML [JBR99] qui ne sont que graphiques et donc seulement échangeable entre êtres humains, le fait, pour EXPRESS, de posséder à la fois une version graphique et une version textuelle rend ce langage traitable par machine. Un modèle EXPRESS peut être échangé entre machines différentes. Il peut également être traité par machine pour en vérifier la cohérence, ou pour passer automatiquement de la version graphique à la version textuelle et inversement. Il peut enfin être exploité pour générer automatiquement différents programmes ou représentations informatives tels que le schéma d'une base de données susceptible d'en stocker les instances. Il s'agit donc bien à la fois d'un modèle interprétable par l'utilisateur humain, et d'une spécification interprétable par machine.

Comparé aux autres notations existantes, EXPRESS possède les points forts suivants :

- il s'agit d'un standard international stable, bien documenté et de plus en plus utilisé ;
- il possède une syntaxe et une sémantique précises autorisant, avec des outils appropriés, une interprétation et un traitement automatique des modèles ;
- il possède une représentation graphique qui fournit une vue synthétique du modèle pour mieux le comprendre ou le concevoir ;
- il possède un système de contraintes complet (contraintes de cardinalité, contraintes ensemblistes, contraintes assertionnelles, contraintes fonctionnelles, contraintes sur les classes...), ainsi qu'un langage procédural permettant d'enrichir celles-ci ;

enfin il permet la ré-utilisation, grâce à sa modularité, des schémas de ressources existants et autorise donc un développement incrémental de modèles de plus en plus vastes et complexes.

De plus, ainsi que nous allons le voir à la section suivante, EXPRESS est associé à des méthodes d'échange et d'accès aux données, et à des outils de génération automatique permettant de générer des programmes de traitement conformes à ces méthodes d'échange et d'accès. EXPRESS constitue ainsi une nouvelle approche puissante et rigoureuse pour la modélisation, l'échange et l'accès aux données.

## 2.3 Méthodes de mise en œuvre

Ce sont les méthodes, prédéfinies et normalisées, de représentation d'instances conformes à un modèle EXPRESS. Ces méthodes de mise en œuvre vont permettre à la fois l'échange de données entre systèmes hétérogènes, et l'archivage de données dans un format indépendant du système source.

### 2.3.1 Représentation des instances sous forme de fichier : le fichier neutre STEP

Ce mode de représentation des instances d'un modèle défini en EXPRESS est spécifié dans le fascicule 21 de STEP [ISO13030.21-94]. Cette spécification définit la manière de coder sous forme d'un fichier de caractères, une population d'instances d'entités conformes à un modèle défini en EXPRESS.

Ce fascicule spécifie :

- les alphabets utilisables et le codage des différentes structures lexicales,
- la structure des fichiers,
- le contenu des entêtes de fichiers,
- les règles de traduction de définitions en EXPRESS,
- le format physique des fichiers.

Un exemple commenté d'instances du modèle de données que nous avons décrit Figure 4 est présenté ci-après :

```
ISO-10303-21;

HEADER;
FILE_DESCRIPTION(('ceci est un test','le fichier contient la description de ...'),'1')
FILE_NAME('nom','1999-08T15:12:30','E. Sardet et G. Pierra',
'LISI/ENSMA','preproc_version','systeme','autoris');
FILE_SCHEMA('etablissement_schema');
END_SEC;

DATA;
#1 = ETUDIANT ('1700975121457', /* num_ss, hérité de la classe personne */
'Dupont', /* nom, hérité de la classe personne */
(15.5, 12.0, $, $, $), /* notes, le tableau de réels optionnels*/
(#5, #6, $, $, $)); /* suit, référence aux instances de cours correspondants*/
#2 = ETUDIANT ('2700286054018', 'Durand', (5.0, 19.0, $, $, $), (#5, #7, $, $, $));
#3 = ENSEIGNANT ('1541211100004', 'Martin', $, (#5, #6));
#4 = ENSEIGNANT ('1600366015259', 'Dupont', $, (#7));
#5 = COURS(.MATH.);
#6 = COURS(.INFO.);
#7 = COURS(.SPORT.);
..
END_SEC;

END-ISO-10303-21;
```

Figure 6 – Un exemple de fichier physique

On peut noter que chacune des instances est identifiée par un *oid* (object identifier: #i), est caractérisée par le nom de la classe instanciée, et contient la suite de valeurs de ses attributs entre parenthèses. Les collections de valeur sont également présentées entre parenthèses, la valeur "indeterminate" (des attributs optionnels) est désignée par '\$', et les valeurs des types énumérés par une notation entourant l'identificateur par un point à chacune de ses extrémités. On peut aussi remarquer que les attributs dérivés et inverses ne sont pas représentés. En effet, lors d'un échange de ce fichier d'instances, le système receveur basé sur le même modèle de données pourra recalculer ces attributs. Un tel fichier permet à deux systèmes informatique d'échanger sans aucune ambiguïté des populations d'instances conformes à un modèle EXPRESS.

### 2.3.2 Interface normalisée d'accès aux instances d'un modèle EXPRESS: l'interface SDAI

En plus de la méthode d'échange de données définie à travers la notion de fichier neutre, une méthode de partage des instances d'un modèle EXPRESS entre systèmes hétérogènes a également été définie. Celle-ci est basée sur la définition d'une bibliothèque de fonctions d'accès normalisées (une interface d'accès), et est connue sous le nom de SDAI (Standard Data Access Interface) [ISO10303.22-97].

Cette interface permet en particulier :

- l'accès et la manipulation par programme d'instances d'entités définies en EXPRESS ;
- l'accès simultané à plusieurs bases de données par plusieurs applications ;
- l'accès à la définition EXPRESS des éléments de données qui peuvent être manipulés par une application ;
- la vérification des contraintes définies dans le modèle de données EXPRESS.

La spécification décrite dans le fascicule 22 définissant l'interface d'accès normalisée indépendamment de tout langage de programmation, des syntaxes d'appel dans des langages de programmation tels C, C++ ou JAVA ont également été définie dans les fascicules 23, 24 et 27. Les applications souhaitant accéder à des modèles de données EXPRESS, et manipuler des instances relatives à ceux-ci, font appel aux services fournis par ces interfaces. Elles peuvent alors s'exécuter sur n'importe quel système supportant la même interface normalisée.

### 2.3.3 Génération d'applications à partir d'un modèle EXPRESS

De nombreux outils existants, gratuits ou commercialisés sur le marché (voir : <http://www.nist.gov/sc4/tools/express/>), permettent de générer automatiquement à partir d'un modèle EXPRESS :

- une base de données susceptible de stocker les instances de ce modèle,
- une interface normalisée (SDAI) d'accès à cette base de donnée,
- un programme de lecture d'un fichier neutre conforme à ce modèle et capable de stocker son contenu dans la base de données par appel de la SDAI.
- un programme d'écriture d'un fichier neutre conforme à ce modèle et capable de lire son contenu dans la base de données par appel de la SDAI, puis de générer le fichier.

Il s'agit donc d'une véritable application générée de façon complètement automatique. Ceci montre l'intérêt considérable du langage EXPRESS, en particulier lorsqu'on le compare aux notations graphiques telles que OMT ou UML.

## 2.4 Les Modèles de données

Le projet STEP n'a pas seulement défini des outils et méthodes, il a également, dans un nombre croissant de domaines, défini des modèles.

### 2.4.1 Architecture générale

L'approche suivie a consisté, d'abord, à définir la structure générale d'une modélisation de produit devant être respectée dans tous les domaines d'application particuliers. Un produit ("product") peut posséder plusieurs variantes (plusieurs "product\_definition\_formation" référencent le même "product"), chacune décrite par plusieurs vues, chacune associée à plusieurs propriétés, ayant plusieurs représentations éventuelles.

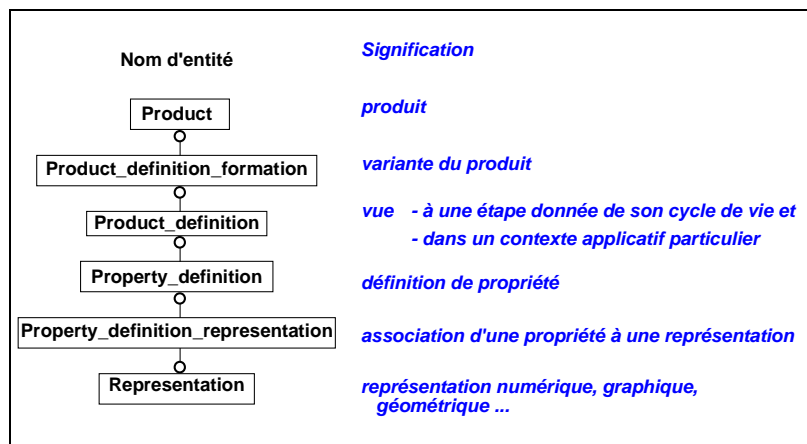


Figure 7 – Structure générale des modèles STEP

Notons que, dans la plupart des domaines, ce sont surtout des propriétés très génériques qui ont été définies et utilisées par STEP : la forme (géométrie), l'identification, le matériau constituant, etc. Les propriétés techniques spécifiques de chaque objet (le diamètre fileté d'une vis, la tension de rupture d'une diode, ...) ne sont en général pas représentées. C'est précisément un des domaines que vise à couvrir P-LIB.

### 2.4.2 Les ressources intégrées

La deuxième étape a consisté à identifier les sujets d'intérêts communs à plusieurs domaines, à abstraire les caractéristiques communes, et à définir des schémas "ressources" susceptibles d'être ensuite spécialisés dans chaque domaine d'application. Les fascicules de la série 40 de STEP décrivent les modèles à vocation très générale. Les fascicules de la série 100 décrivent les modèles moins universels car correspondant à des grands secteurs applicatifs. Les modèles figurant dans ces différents fascicules sont cohérents entre eux et non redondant. On les qualifie de "ressources intégrées".



Parties	Titres	Situation
Partie 41	Description générique des produits	IS
Partie 42	Géométrie, topologie et représentation des formes	IS
Partie 43	Structures de représentation	IS
Partie 44	Configuration des produits	IS
Partie 45	Materials	IS
Partie 47	Tolerances	IS
Partie 49	Process structure	IS
Partie 101	Dessin technique	IS
Partie 104	Finite Element Analysis	DIS
Partie 105	Kinematics	IS
Partie 106	Building construction core model	préparation CD

Figure 8 – Les ressources intégrées de STEP

Il est intéressant de noter le caractère très générique des ressources ainsi construites. Par exemple, chaque entreprise est confrontée, lorsqu'elle déploie un système de gestion de données techniques (SGDT) [RAN95], à définir sa propre structure de modélisation de produit [FV98]. Chaque bureau de calcul doit définir sa propre organisation des données de calcul par éléments finis. Les modèles génériques de STEP, élaborés par des spécialistes de chacun de ces domaines, contiennent une proposition de solution pour chacun de ces problèmes [FV98].

### 2.4.3 Les protocoles d'application STEP

Mais les approches génériques ne suffisent pas. La caractéristique de chaque métier est d'avoir élaboré son propre savoir faire par spécialisation du savoir faire commun aux différents métiers techniques. Un ingénieur de calcul n'utilise pas le même langage qu'un responsable de production, de même qu'un électricien ne s'intéresse pas aux mêmes objets qu'un plombier. C'est à ce niveau de spécialisation -un type d'activité (l'électronique), une phase du cycle de vie (la conception) et un métier (la schématique)- que les données réelles doivent être conservées, et si possible articulées avec les niveaux voisins. C'est, dans STEP, le niveau des protocoles d'applications.

Les spécialistes du domaine ont la charge de définir les activités propres à leurs métiers (c'est l'AAM ou Application Activity Model), les grandes catégories d'objets ou de concepts mis en jeu (ce sont les unités fonctionnelles, ou Unit of Functionality, UoF) et finalement, mais de façon plus ou moins précise et formelle, les données nécessaires et les contraintes qu'elles doivent respecter (c'est le modèle de référence ou Application Référence Model ARM). Cet ensemble de modèles, décrit par les spécialistes du domaine, correspond à une spécification de besoins.

Une coordination est alors organisée avec des spécialistes de la modélisation de données et de l'architecture STEP, de façon à exprimer les besoins identifiés dans le modèle de référence (ARM) à l'aide des seules entités figurant dans les ressources intégrées de STEP, éventuellement combinées par héritage. Ce processus est appelé une "interprétation" des besoins à l'aide des ressources communes. Le résultat, ou AIM (Application Interpreted Model), est supposé traduire l'ensemble des besoins identifiés par les spécialistes du domaine en termes des seules ressources intégrées de STEP. La figure ci-dessous montre le principe de réalisation d'un protocole d'application.

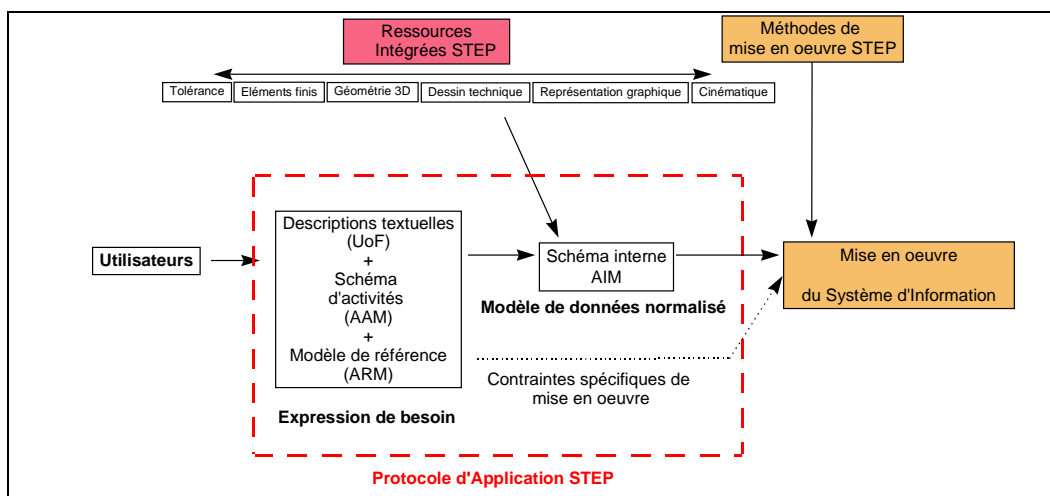


Figure 9 – Principe d'un protocole d'application [FV98]

Initialement, l'ARM pouvait être exprimé de façon textuelle et à l'aide de notations graphiques telle que NIAM [HAB93] ou IDEF1x[BOU95a]. Une phase d'interprétation était alors nécessaire pour réécrire ce modèle en EXPRESS. Au fur et à mesure

que la technologie EXPRESS diffuse, les modèles de référence sont de plus en plus souvent écrits directement en EXPRESS par les experts du domaine. Ceci rend discutable la nécessité du processus d'interprétation, d'autant que celui-ci, assez informel, n'assure pas qu'un même besoin d'information sera interprété de la même façon (i.e., représenté par les mêmes entités) dans deux protocoles d'application différents. Des évolutions sont donc en cours pour essayer de modulariser les protocoles d'application de façon à promouvoir la réutilisabilité à un niveau plus élevé que les simples entités EXPRESS.

La figure ci-après liste l'ensemble des protocoles d'applications, développés ou en cours, et montre la très grande variété des domaines applicatifs couverts.

Parties	Titres	Situation
Partie 201	Explicit draughting	IS (1994)
Partie 202	Associative draughting	IS (1996)
<b>Partie 203</b>	<b>Configuration controlled design</b>	<b>IS (1994)</b>
Partie 204	Mecha. design using B Rep	préparation FDIS
Partie 205	Mecha. design using surface rep	DIS
Partie 207	Sheet metal die planning and design	preparation FDIS
Partie 208	Product life cycle support	préparation CD
Partie 209	Design through analysis of composite	préparation DIS
Partie 210	Electronic PCA: design and manufacture	préparation DIS
Partie 212	Electrotechnical plants	vote CD
Partie 213	NC process plans for machined parts	préparation FDIS
<b>Partie 214</b>	<b>Core Data for automotive mech. design</b>	<b>préparation FDIS</b>
Partie 215	Ship arrangement	préparation CD
Partie 216	Ship moulded forms	préparation CD
Partie 217	Ship piping	préparation CD
Partie 218	Ship structures	préparation CD
Partie 221	Functional data & schematics for process plants	préparation CD
Partie 222	Design manufacturing for composite structures	préparation CD
Partie 223	Design and manufacturing info for cast parts	préparation CD
Partie 224	Product definition for process planning	préparation DIS
Partie 225	Structural building elements	vote DIS
Partie 227	Plant spatial configuration	préparation FDIS
Partie 230	Buiding structural frame: steelwork	preparation CD
Partie 231	Process engineering data	préparation CD
Partie 232	Technical data package	préparation CD

Figure 10 – Les protocoles d'application en cours

Les approches suivies dans les différents protocoles d'application sont, en général, assez largement différentes. L'intégration de domaine à domaine laisse encore ouvertes un grand nombre de questions. Néanmoins, dans chaque domaine applicatif, le modèle de référence (ARM) d'un protocole d'application représente une véritable capitalisation du savoir-faire d'un grand nombre d'experts du domaine sur les informations qui sont pertinentes dans ce domaine, et sur la structuration qu'il convient de leur donner.

Dans le domaine mécanique, les protocoles d'applications 203 et 214 sont assez représentatifs de cet apport. Le 203 [FV98] définit un modèle de données pour la mise en œuvre du Système de Gestion de Données Techniques couplé à des systèmes de CAO dans les industries mécaniques. Plus ambitieux encore, le 214, élaboré par l'ensemble des fabricants automobiles du monde entier, propose une définition des données nécessaires pour représenter des produits aussi complexes que les véhicules automobiles tout au long de leur cycle de vie.

## 2.5 Les tests de conformité

Instruites par les difficultés rencontrées dans les interfaces de la génération précédente (telle que IGES) où il était très difficile, en cas d'anomalie dans un transfert de données, d'identifier les responsabilités entre l'émetteur, le receveur, voire les erreurs ou imprécisions de la spécification elle-même, la norme STEP définit pour chaque AP :

- Comment doivent être testées les interfaces ?
- Sur quels critères établir leur conformité ?

A l'ISO, des projets de normes définissant les tests de référence sont en cours de développement pour les AP202, 203, 204. D'autres tests de référence doivent être progressivement développés pour tous les protocoles d'application.

En France, une Norme expérimentale pour l'AP 203 (AFNOR Z 68 333), un Laboratoire de Tests au GOSET, et un Comité de Certification NF à l'AFNOR sont déjà disponibles. Certaines interfaces sont d'ores et déjà certifiées conforme pour ce protocole d'application dont le déploiement est actuellement le plus avancé.

## 2.6 De la modélisation de produit à la caractérisation des composants

STEP a donc développé toute une nouvelle technologie pour permettre la modélisation informatique des données des produits, tels que ceux-ci existent dans l'entreprise. C'est la notion de maquette numérique pour laquelle STEP définit une représentation neutre, c'est-à-dire indépendante des outils logiciels qui la génèrent ou l'exploitent.

Dans une telle représentation, chaque produit et chaque composant seront décrits par ce qu'ils sont (une forme, une matière, une identification) mais non par les raisons pour lesquelles ils sont là. Pourquoi tel composant a-t-il été choisi plutôt que tel autre ? Quels ont été les critères ou les propriétés qui ont amené à le choisir ? Comment conviendrait-il de le remplacer si sa référence n'était plus fabriquée ? Ceci constitue des exemples de questions auxquelles la représentation STEP ne fournit pas de réponse. Il faudrait pour cela identifier les propriétés qui ont été déterminantes lors du choix de chaque composant. Il faudrait connaître les exigences exprimées par le concepteur sur les valeurs de ces propriétés. Il faudrait enfin modéliser et enregistrer ces exigences au sein de la maquette numérique. C'est le complément que P-LIB va apporter à STEP lorsqu'un produit comporte, en particulier, des composants préexistants.

## 3. Modélisation et échange de données de composants : Le projet P-LIB

Démarré en 1987 au niveau Européen, puis en 1990 au niveau ISO, la norme ISO 13584 (Parts Library, connue en Europe sous le nom de CAD/LIB, [PIE94] et désormais associée à l'abréviation P-LIB) est la norme destinée à permettre la modélisation, l'échange et le référencement à partir de STEP, de bibliothèques informatisées de composants préexistants. P-LIB permet aussi bien la représentation de composants abstraits, tels qu'ils sont par exemple utilisés lors d'une conception fonctionnelle (un système de pompage, une alimentation, une liaison rotule) que celle de composants fournisseurs ou normalisés (une vis CHc, M 12-35, NF E 27-161, un palier-applique à billes INA KGHWT 25 PP). Cette norme permet également d'associer à un objet bibliothèque, un nombre quelconque de représentations, propres à chacune des disciplines qui manipule l'objet (une vue solide pour la conception mécanique, une vue symbolique pour la conception fonctionnelle, un modèle de comportement pour la simulation analogique, etc.). Elle permet enfin d'intégrer dans un environnement homogène et cohérent des bibliothèques fournies par différentes sources [PIE97].

L'idée centrale de la norme P-LIB est qu'un composant n'est pas seulement, comme le considère les systèmes CAO, une ou plusieurs *représentations*. Ce n'est pas non plus, comme le considère les bases articles ou, bien souvent, les fichiers STEP, une simple *identification*. C'est, en fait, un ensemble de propriétés spécifique de chaque famille d'objet, et qui représente aussi bien la nature que la fonction des composants de la famille. C'est autour de la définition, de la représentation, et de l'exploitation de la notion de propriété qu'est construite la plus grande partie de la norme P-LIB.

La structure de la norme P-LIB est voisine de celle de STEP : sept parties (ISO 13584-1, 10, 20, 24, 26, 31 et 42) constituent le noyau de la spécification et sont communes à toutes les disciplines. La série 100 (ISO 13584-1xx), appelée "view exchange protocols", est prévue pour accueillir les spécifications des représentations propres à chaque secteur d'activité au fur et à mesure de leur développement.

P-LIB couvre deux types de problèmes :

- Identifier et représenter les propriétés et les familles d'objets qui ont "la même signification" : c'est la notion de dictionnaires de données,
- Saisir, modéliser et échanger la *connaissance* sur des composants : c'est la notion de bibliothèques de composants, dont la représentation sous forme de document aboutit à des catalogues informatisés "actifs" ou "intelligents".

### 3.1 Définition de dictionnaires de données pour le domaine des objets techniques

Echanger ou référencer des bibliothèques nécessite d'abord de pouvoir identifier, et décrire sous forme informatique, les catégories d'objets (vis, capacité ou pompe) et les types de propriétés (diamètre fileté, tension de rupture ou température maximale supportée) concernées. C'est la notion de dictionnaire de données.

La spécification correspondante, dont le domaine d'application déborde largement le problème des bibliothèques de composants puisqu'elle permet d'informatiser tant les normes de terminologie que les attributs techniques propres aux produits d'une entreprise, a été développée conjointement avec la CEI (Commission Electrotechnique Internationale). Ceci assure un traitement homogène de tous les types de composants. Cette spécification constitue la partie 42 de l'ISO 13584, déjà publiée en tant que norme internationale (IS : International Standard). Elle est également publiée par la CEI sous la référence IEC 61630-2.

#### 3.1.1 Principes fondamentaux de la modélisation

Cette spécification est basée sur deux principes de bases :

- 1) les différentes catégories d'objets doivent, conformément à l'approche objet [MEY98] [CY92], être organisées selon une hiérarchie de "familles" ou de "classes", avec factorisation/héritage des propriétés.
- 2) hiérarchie de classes et propriétés doivent être définies simultanément :
  - les propriétés applicables précisent la signification d'une classe de composants ;
  - le domaine d'application précise la signification d'une propriété.

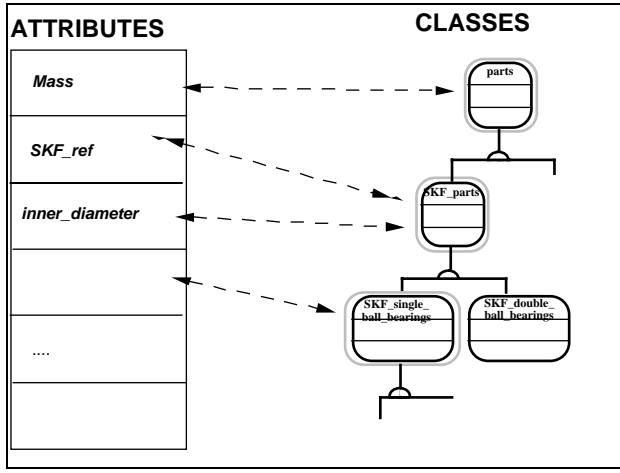


Figure 11 - Définition simultanée des classes et propriétés

Dans l'exemple ci-dessus, la notion de "masse" est associée à la classe "parts" (composant mécanique). Cette propriété précise le concept de "composant mécanique": ce sont des composants pour lesquels la masse est définie (à la différence, par exemple, des composants logiciels). Cette propriété est également définie implicitement pour toutes les sous-classes de composants mécaniques (c'est ce que l'on appelle l'"héritage"). Au contraire le "diamètre\_intérieur\_de\_roulement" est seulement défini pour l'ensemble des composants d'un catalogue SKF de roulements, mais pas pour les autres sous-arbres qui pourraient contenir des vis, vérins, transistors, etc.

### 3.1.2 Typologie des propriétés

Lorsque l'on étudie les propriétés d'un composant, les premières propriétés, évidentes, sont celles qui caractérisent sa forme ou sa fonction. Le diamètre intérieur, ou la vitesse de rotation maximale d'un roulement par exemple. Mais pour choisir un roulement, ce qui intéresse surtout l'utilisateur est sa durée de vie. Et celle-ci est elle-même fonction des conditions d'utilisation : vitesse de rotation effective, force radiale subie, etc. Ainsi P-LIB définit trois catégories de propriétés illustrées dans la figure ci-dessous.

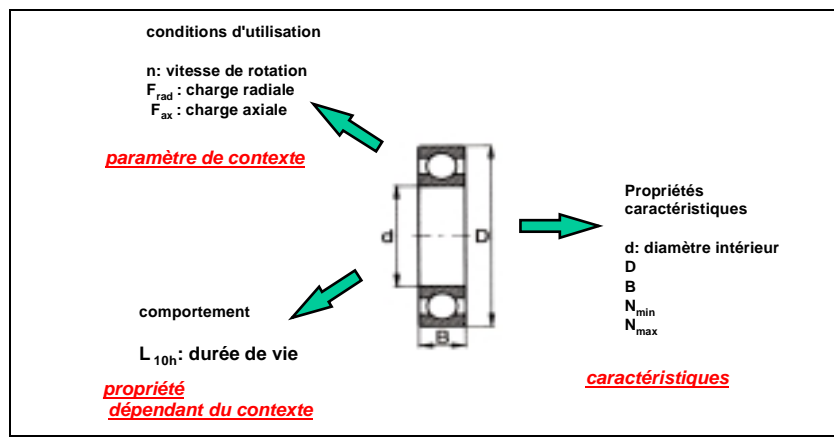


Figure 12 - Typologie des propriétés

Les paramètres de contexte, et les valeurs requises pour les propriétés dépendant du contexte, permettent à l'utilisateur de définir le problème qu'il a à résoudre. La description du comportement, c'est-à-dire la représentation des méthodes de calcul des propriétés dépendant du contexte en fonction des caractéristiques du composant et des paramètres de contexte permettent au fournisseur de modéliser et d'échanger la connaissance qu'il possède sur le comportement de ses composants.

### 3.1.3 Méthode de description

A partir de cette typologie, P-LIB fournit un modèle de données (et un format de description documentaire dans la norme sœur [IEC 61630.4-97]), pour la hiérarchie classe-propriété. On remarquera que chaque classe ou chaque propriété est associée à un code qui permettra d'identifier de façon universelle tout élément d'un dictionnaire P-LIB (cf. 3.1.4).

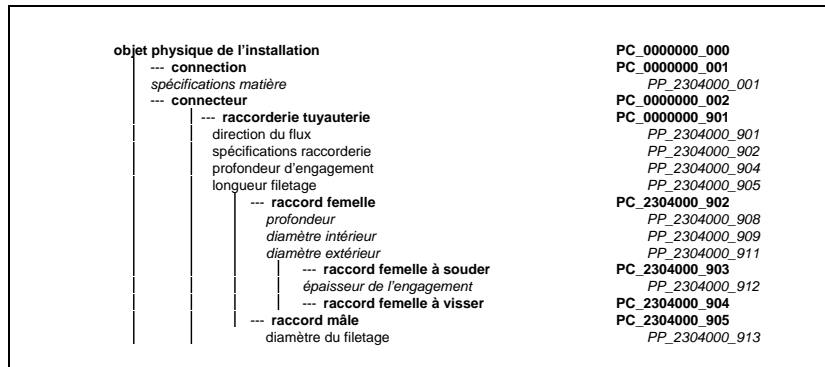


Figure 13 - Exemple de hiérarchie Classes/propriété  
(proposée pour le protocole d'ingénierie de STEP )

P-LIB fournit également un modèle de donnée pour décrire avec beaucoup de précision chacune des propriétés et chacune des classes.

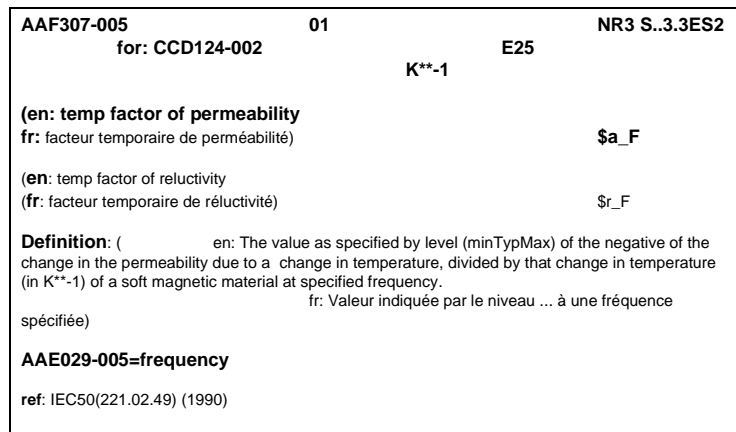


Figure 14 - Exemple de définition dictionnaire d'une propriété [IEC 61630.4-97]

L'exemple ci-dessus, tiré de la norme CEI 61360-4, montre les différentes informations qui doivent être associées à la description d'une propriété dans un dictionnaire conforme à P-LIB. Outre le code de la propriété (AAF307-005), et de la classe pour laquelle elle est définie (CCD124-002), on trouve l'unité, la dimension, le format d'échange de valeur par défaut, le nom, la définition, les références et, dans le cas des propriétés dépendant du contexte, les paramètres de contexte dont elles dépendent (ici AAE 029-005 : la fréquence du courant).

### 3.1.4 Méthode d'identification

P-LIB offre également une identification non ambiguë pour n'importe quelle propriété définie dans un dictionnaire P-LIB : la concaténation du code de l'entité (norme ou société) ayant défini la classe, du code de la classe (unique pour cette entité), et du code de la propriété (unique pour cette classe) définit en effet de façon universelle et non ambiguë chaque propriété. C'est ce simple code, appelé un BSU (Basic Semantic Unit), qu'il sera suffisant de mémoriser dans un modèle de produit (par exemple un fichier STEP) pour caractériser une propriété.



Figure 15 - Identification du facteur de perméabilité d'un matériau magnétique à une fréquence donnée  
(le code 0112/2///61630-4 caractérise en effet la norme CEI 61360-4)

### 3.1.5 Un exemple

L'exemple ci-après montre le dictionnaire de données, classes et propriétés, défini par un fabricant de roulements pour informatiser son catalogue conformément à la norme P-LIB.

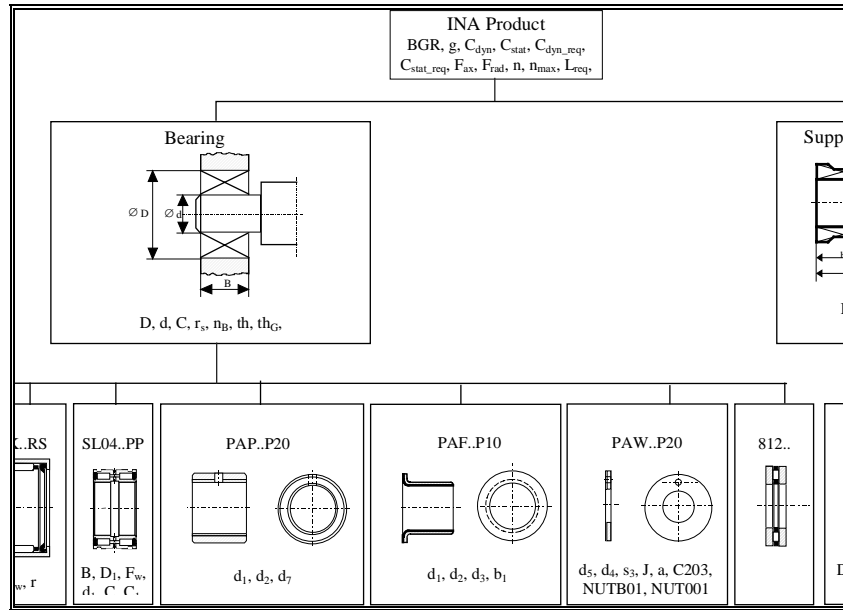


Figure 16 - Un exemple de dictionnaire P-LIB dans un catalogue fournisseur

Les propriétés définies à chacun des niveaux sont implicitement définies pour tous les niveaux inférieurs et peuvent être utilisées pour des requêtes à chacun des niveaux où elles sont définies.

### 3.2 Des dictionnaires de données aux bibliothèques de composants

Si les dictionnaires permettent de définir les familles de composants et les types de propriété qui leur sont associés, ils ne permettent ni de sélectionner un composant particulier dans une famille en calculant toutes les valeurs de ses propriétés (par exemple : calculer la durée de vie d'un roulement en fonction de la charge dynamique que l'on va lui appliquer et de sa vitesse prévue de rotation), ni d'en générer des représentations (par exemple création de la représentation géométrique à partir d'un programme paramétré). Ce domaine correspond à la notion de bibliothèque à proprement parler. Il est couvert par les parties 20 (publiée comme Standard International : IS) et 24 (DIS en cours de vote FDIS) de P-LIB. Les modèles de données définis dans ces spécifications permettent à la fois de représenter les domaines de valeurs autorisés pour les propriétés susceptibles d'être choisies par l'utilisateur (propriétés caractéristiques et paramètres de contexte), et de définir les méthodes de calcul ("fonctions de dérivation") pour les propriétés pouvant être calculées. La figure ci-dessous montre l'ensemble des informations susceptibles d'être fournies par un fabricant de roulements:

- la table de caractéristiques, dont le contenu est fournie par le fabricant de roulements, décrit tous les roulements de la famille;
- les paramètres de contexte sont définis par le fabricant de roulements mais leurs valeurs sont destinées à être fournies par l'utilisateur, enfin
- les propriétés dépendant du contexte sont destinées à être automatiquement calculé par le système de gestion (le "bibliothécaire") à partir, d'une part, des valeurs fournies par l'utilisateur pour les caractéristiques choisies et les paramètres de contexte, et, d'autre part, des fonctions de dérivation fournies par le fabricant.

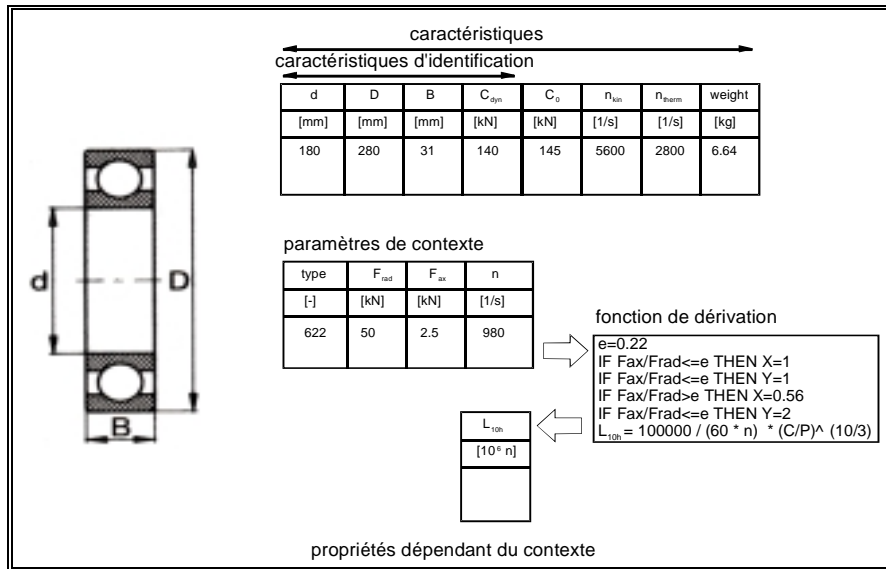


Figure 17 - Connaissance modélisée sur une famille de roulements

Grâce à ces informations, l'utilisateur pourra, par exemple, chercher tous les roulements ayant des diamètres intérieurs et extérieurs fixés, et ayant, dans des conditions données de charge axiale, de charge radiale et de vitesse, une durée de vie supérieur à un certain seuil. Le système bibliothécaire sera alors capable, en exploitant les seules informations fournies par le fabricant, de calculer le ou les roulements satisfaisant aux exigences énoncées.

Notons que les caractéristiques d'identification (qui apparaissent dans la figure ci-dessus) sont les seules caractéristiques qu'il est indispensable de conserver pour identifier le composant au sein de sa famille.

### 3.2.1 Représentations

L'architecture de la norme P-LIB permet alors au fournisseur d'associer autant de représentations que nécessaire aux composants de la bibliothèque. La définition du composant, qui correspond à ce que nous avons présenté à la section précédente, est définie dans une hiérarchie de classes dite "classes de modèles généraux". Chaque représentation est définie par une autre hiérarchie de classes dite "classes de modèles fonctionnels" [PIE94].

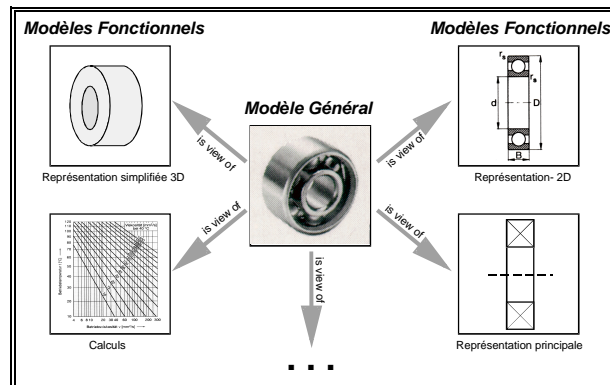


Figure 18 - Connaissance modélisée sur une famille de roulements

Chaque nouvelle classe de modèles fonctionnels est définie par un fascicule de la série 100. Actuellement deux tels fascicules existent. Le fascicule 101 (view exchange protocol by parametric representation) permet de fournir des représentations géométriques sous forme de programmes de géométrie paramétrée. Le fascicule 102 (view exchange protocol by ISO 10303 conforming representation) permet d'associer à chaque composant de la bibliothèque n'importe quelle représentation susceptible d'être définie par un protocole d'application de STEP.

### 3.3 Des bibliothèques de composants aux catalogues informatisés intelligents

Ainsi que nous l'avons montré précédemment, un modèle de données EXPRESS peut permettre de saisir toute la connaissance sur le comportement et les critères de sélection d'un composant. Mais le fichier d'échange résultant (cf. figure 6) est extrêmement peu lisible.

Or, il existe également un autre manière de présenter une connaissance sur un univers de composants : c'est la méthode documentaire telle qu'elle est utilisée dans les catalogues papiers, ou telle qu'elle commence à être utilisée dans les documents électroniques de type pdf, HTML ou XML.

La confrontation EXPRESS-HTML/XML donne la comparaison suivante :

	EXPRESS	HTML/XML
+	<ul style="list-style-type: none"> <li>• <b>cohérence</b> : contraintes d'intégrité (sémantique des données)</li> <li>• <b>connaissance modélisée de façon explicite</b> : utilisable par machine (sélection...)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>très synthétique</b> et donc compréhensible par l'homme</li> <li>• peut devenir "<b>dynamique</b>" (liens, applets Java, ...)</li> </ul>
-	<ul style="list-style-type: none"> <li>• <b>fichier d'instances pas lisible pour l'homme</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>pas de garantie de cohérence</b> : pas de contraintes d'intégrité (possibilité d'erreur de contenu)</li> <li>• <b>connaissance implicite</b> : inutilisable par machine</li> </ul>

Il était donc tentant d'essayer de marier les deux univers pour fournir une représentation et une interface documentaire, et donc lisible (un catalogue), à la connaissance explicite, donc traitable par machine, représentée en P-LIB (domaine de valeur, fonction de dérivation, ...)

Ce travail a été fait récemment [SAR99] à travers :

- une méthode permettant d'associer à un modèle de données EXPRESS une structure de document XML (une "DTD") permettant de présenter de façon lisible la même information ;
- un environnement de développement permettant d'associer à un document XML la représentation explicite de l'information qu'il contient sous forme d'une population EXPRESS.

Cet environnement permet alors, à partir d'un document purement statique de type HTML (susceptible d'être généré, par exemple, à partir d'Acrobat™ ou de Word™) :

- de rajouter interactivement, au sein d'un éditeur SGML/XML, des balises représentant des concepts P-LIB,
- de vérifier, pour chaque information balisée, qu'elle respecte les contraintes d'un modèle de données EXPRESS,
- de générer automatiquement un fichier neutre d'échange STEP tel que défini dans la Figure 6,
- de générer automatiquement un document "actif" où les tables, par exemple, ont été remplacées automatiquement par des programmes (des "applets Java") permettant à l'utilisateur d'interagir avec le document, par exemple pour sélectionner une valeur dans la table.

La figure suivante montre la chaîne de traitement réalisée.

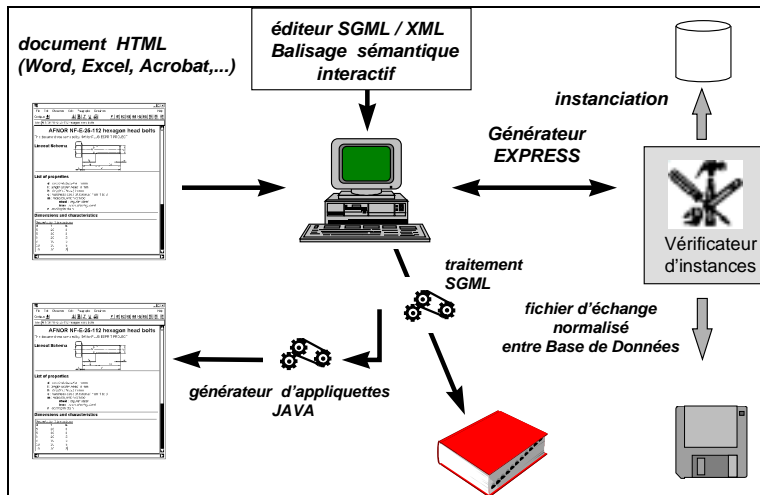


Figure 19 - Une chaîne complète de traitement [SAR99]

Cette chaîne de traitement génère, à partir d'un simple catalogue de type HTML ou pdf, à la fois un fichier d'échange P-LIB, un catalogue informatisé "actif" pour l'Internet et, par un traitement de mise en forme spécifique en SGML, le programme de commande d'une photocomposeuse pour l'impression papier. La mise au point de cette chaîne a été réalisée dans le cadre d'un partenariat entre EDF, TOSHIBA Corp., ITEMATIC, le LISI/ENSMA et AUBIN Imprimeur.

#### 4. Représentation de composants P-LIB dans un modèle STEP

Si les représentations (par exemple géométriques) générées par une bibliothèque P-LIB doivent évidemment pouvoir être échangées dans un modèle STEP, deux autres besoins ont progressivement émergés. D'une part, un besoin de traçabilité, visant



à représenter dans STEP non seulement l'objet bibliothèque, mais également la référence à sa source dans une bibliothèque P-LIB, et les critères de choix qui ont abouti à sélectionner cet objet. D'autre part, un besoin d'intégration plus profonde des deux normes visant à permettre dans un modèle STEP le seul échange des références à P-LIB, le système receveur régénérant, si besoin est, les représentations à partir de sa propre bibliothèque.

Deux ans se sont avérés nécessaires à la fois pour développer une solution technique et dégager un consensus. A l'heure actuelle, des modèles de données EXPRESS existent pour satisfaire ces deux besoins au sein d'un protocole d'application [STA98].

La figure suivante montre l'infrastructure informatique retenue dans certaines grandes entreprises d'ingénierie pour gérer l'ensemble de leurs données techniques. Le référentiel produit ne contient des composants que les seules informations nécessaires pour les identifier ou enregistrer les raisons qui ont conduit à les sélectionner. Le référentiel composant quant à lui décrit exhaustivement, en format P-LIB, les composants dont l'usage est, ou a été, autorisé dans l'entreprise. Il permet également la recherche de composants adaptés lors des phases de conception.

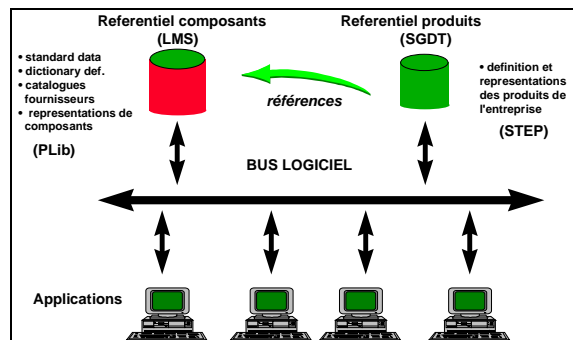


Figure 20 - Gestion intégrée des produits et composants

Une telle architecture permet:

- d'éviter la redondance,
- de tracer le pourquoi des choix, enfin
- de faciliter la maintenance ultérieure des produits (usines ou installations).

## 5. Conclusion

La généralisation de l'informatique dans le domaine technique impose à chaque entreprise de choisir les langages, les modèles, les outils et les systèmes lui permettant de représenter, d'exploiter et d'archiver son patrimoine informationnel sous forme numérique. Dans cette conférence, nous avons montré les multiples apports résultant des projets STEP et P-LIB pour faciliter ces choix.

Au niveau langage, le langage EXPRESS apparaît comme le formalisme de spécification le plus avancé aujourd'hui pour la représentation des données hautement structurées telles qu'on les rencontre dans la modélisation d'objets techniques, par exemple dans le domaine de la géométrie. Il permet d'exprimer n'importe quel type de contraintes. Il spécifie à la fois des formats d'échange et d'archivage, et des interfaces d'accès. Enfin son caractère textuel le rend compilable : programmes de lecture de fichiers d'échanges et interface d'accès à des bases de données peuvent être générés automatiquement à partir du seul modèle de données.

Au niveau des modèles de données, STEP définit à la fois des ressources génériques pour la modélisation des produits, et des modèles complets et finalisés pour un certain nombre de domaines. P-LIB définit des ressources génériques pour la modélisation du *comportement* des produits et composants. Il définit un modèle et une méthode pour fabriquer des dictionnaires de types d'objets et de propriétés associés. Il définit enfin un modèle complet et finalisé de catalogues de composants industriels contenant non seulement la description des composants, mais encore la connaissance sur leur comportement et les règles devant présider à leur sélection. L'utilisation conjointe de STEP et de P-LIB dans un modèle de produit permet outre le fait de représenter le produit tel qu'il est, de représenter également, pour les composants qui y figurent, les critères selon lesquels ils ont été choisis et les propriétés qu'ils doivent conserver en cas de substitution ultérieure dans la phase de maintenance. Enfin, des dictionnaires de données basés sur P-LIB existent dès à présent. La norme CEI 61630-4 définit un tel dictionnaire pour le domaine des composants électroniques et des propriétés associées.

Au niveau des outils, les langages et modèles précédents sont instrumentés. Un grand nombre d'outils EXPRESS existent dès à présent, gratuits ou commercialisés, tant sous forme d'éditeur, que de compilateur, de générateur de code ou de gestionnaire de données. Des interfaces STEP apparaissent de plus en plus et commencent à être certifiées. Des gestionnaires de bibliothèques P-LIB, ainsi que des générateurs de catalogues actifs pour l'Internet basés sur le modèle P-LIB, apparaissent en tant que produits commercialisés pour les premiers et de prototypes pour les seconds.

L'ensemble de cette technologie est maintenant stable, mature et largement accepté. Certes elle est encore en phase de déploiement. Les outils EXPRESS peuvent contenir des erreurs, et les modèles de données et interfaces quelques anomalies. Mais c'est le propre de toute nouvelle technologie que de passer par une phase de rodage. Et c'est aussi dans cette phase de rodage que l'on prend un avantage compétitif significatif en maîtrisant une technologie émergente. L'objet de cette conférence était de mettre en évidence les multiples potentialités de cette technologie.

## Remerciements

L'auteur remercie à la fois les ingénieurs de l'Association GOSET, centre de compétences français sur le standard STEP et Eric Sardet, du LISI/ENSMA, pour certains des exemples et figures utilisés dans ce papier et dont il sont à l'origine.

## Bibliographie

- [ABG95] **Ait-Ameur Y., Besnard F., Girard P., Pierra G., Potier J.C.** - *Formal Specification and Metaprogramming in the EXPRESS Language*, In Intern. Conference on Software Engineering and Knowledge Engineering SEKE'95, IEEE - ACM Sigsoft pp. 181-189, Rockville, USA, 1995.
- [APS95] **Ait-Ameur Y, Pierra G., Sardet E.** - *Using the EXPRESS language for metaprogramming*, Proc. of the 3rd Interna. Conf. of EXPRESS User Group EUG'95, Grenoble 21-22 Oct. 1995.
- [ATK 90] **Atkinson M., Bancilhon F., Dittrich K., Maier D., Zoo Nik S.** - *The Object-Oriented Database System Manifesto. Deductive and Object-Oriented Databases.*, Elsevier Science Publishers B.V. (North-Holland), pp. 223-240, 1990.
- [BOU95a] **Bouazza M.** - *La norme STEP*, Hermès, Paris, 1995.
- [BOU95b] **Bouazza M.** - *Le langage EXPRESS*, Hermès, Paris, 1995.
- [BRA98] **Bray T.**, et al, *Extensible Markup Language (XML) 1.0*, WWW Consortium 1998/02/10., <http://www.w3.org/TR/REC-xml>
- [CY92] **Coad P., E. Yourdon** - *Object-Oriented Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1992.
- [FOW] **Fowler J.** - *STEP for data management Exchange and Sharing*. Technology appraisals edition. 1995. ISBN 1-871802-36-9.
- [FV92] **Féru F., Viel Ch.** - *Echanger avec le protocole d'application 203 de STEP. Echange et partage de données CAO et GDT*, Ass. GOSET, Nanterre, ISBN: 2-9513382-0-1, 1998
- [HAB 93] **Habrias H.**, *Introduction à la spécification*, Editions Masson, ISBN: 2-225-82768-0, 1993.
- [IEC61360.4-97] IEC 61360-4 - *Standard data element types with associated classification scheme for electric components — Part 4: IEC reference collection of standard data element types, component classes and terms*, 1997.
- [ISO10303.11-94] ISO 10303-11 - *Industrial automation systems: and integration — Product data representation and exchange — Part 11: Description Methods: The EXPRESS language reference manual*, 1994.
- [ISO10303.21-94] ISO 10303-21 - *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear Text Encoding of the Exchange Structure (Physical File)*, 1994.
- [ISO10303.22-97] ISO 10303-22, *Industrial automation systems and integration — Product data representation and exchange — Part 22: Implementation methods: Standard Data Access Interface*, 1997.
- [ISO13584.20-98] ISO 13584-20 - *Industrial automation systems and integration — Parts library — Part 20: Logical resource: Logical model of expressions*, 1999.
- [ISO 13584.24-98] ISO DIS 13584-24 - *Industrial automation systems and integration — Parts library — Part 24: Logical resource: Logical model of supplier library*, 1999.
- [ISO13584.42-98] ISO 13584-42 - *Industrial Automation Systems and Intregation — Parts Library — Part 42: Description methodology: Methodology for Structuring Parts Families*, ISO, Geneve, 1998.

- [JBR99] **Jacobson I., Booch G and Rumbaugh J.** - *The Unified Software Development Process*, Addison-Wesley Eds, 1999.
- [MEY88] **Meyer B.** - *Object-Oriented Software Construction*, Prentice Hall, Hemel Hempstead, UK, 1988.
- [PIE94] **Pierra G.** - *Modelling classes of pre-existing components in a CIM perspective: the ISO13584/ENV 40014 Approach*, Revue internationale de CFAO et d'Infographie, vol. 9, n°3, 1994, pp. 435-454, 1994.
- [PIE97] **Pierra G.** - *Intelligent Electronic Component Catalogues for Engineering and Manufacturing*, Proc. of the Internat. Symposium on Global Engineering Networking, GEN'97, pp. 331-352, Antwerp, Belgium, April 2-24, ISBN 3-931466-20-5, 1997. .
- [PSP98] **Pierra G., Sardet E., Potier J.C., Battier G., Derouet J. C., Wilman N., Mahir A.** - *Exchange of Component Data: The Plib model, Standard and Tools*, Proc. of the 9<sup>th</sup> Internat. Conf. On Enterprise Integregation and CALS, CALS EUROPE'98, pp. 160-176, Paris, 16-18 Sept., 1998,
- [RAN95] **Randoing J. M.** - *Les SGDT*, Hermès, Paris, 1995.
- [RBP91] **Rumbaugh J., Blaha M., Premerlani W, Eddy F., Lorenzen W.** - *Object Oriented Modelling and Design*, Prentice-Hall International Editions, ISBN 0-13-630054-5, 1991.
- [SG97] **Sardet E., Pierra G.** - *Formal specification, modelling and exchange of classes of components according to P-LIB. A case study*, Proc. of the Internat. Symposium on Global Engineering Networking, GEN'97, pp179 à 201, Antwerp, Belgium, April 2-24, ISBN 3-931466-20-5, 1997
- [SAR99] **Sardet E.** - *Intégration des approches modélisation conceptuelle et structuration documentaire pour la saisie, la représentation, l'échange et l'exploitation d'informations. Application aux catalogues de composants industriels..* Thèse de Doctorat en Informatique de l'Université de Poitiers, Poitiers, 1999
- [STA98] **Staub G.** - *Interpretation of PLIB services: A proposal for interpretation of the "services" provided by PLIB using the STEP Integrated Resources ISO TC184/SC4/QC/N068* ([http://www.nist.gov/sc4/wg\\_qc/qc/qcn068/qcn068.pdf/](http://www.nist.gov/sc4/wg_qc/qc/qcn068/qcn068.pdf/)).
- [SW94] **Schenck D., Wilson P.** - *Information Modelling The EXPRESS Way*, Oxford University Press, 1994.