

**Zeitschrift:** Vermessung, Photogrammetrie, Kulturtechnik : VPK = Mensuration, photogrammétrie, génie rural

**Band:** 81 (1983)

**Heft:** 9

**Artikel:** Recommandations de la commission d'automation de la SSMAF à propos de la représentation des structures logiques de données

**Autor:** Sievers, B.

**DOI:** <https://doi.org/10.5169/seals-231653>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. [Siehe Rechtliche Hinweise.](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. [Voir Informations légales.](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. [See Legal notice.](#)

**Download PDF:** 13.10.2024

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

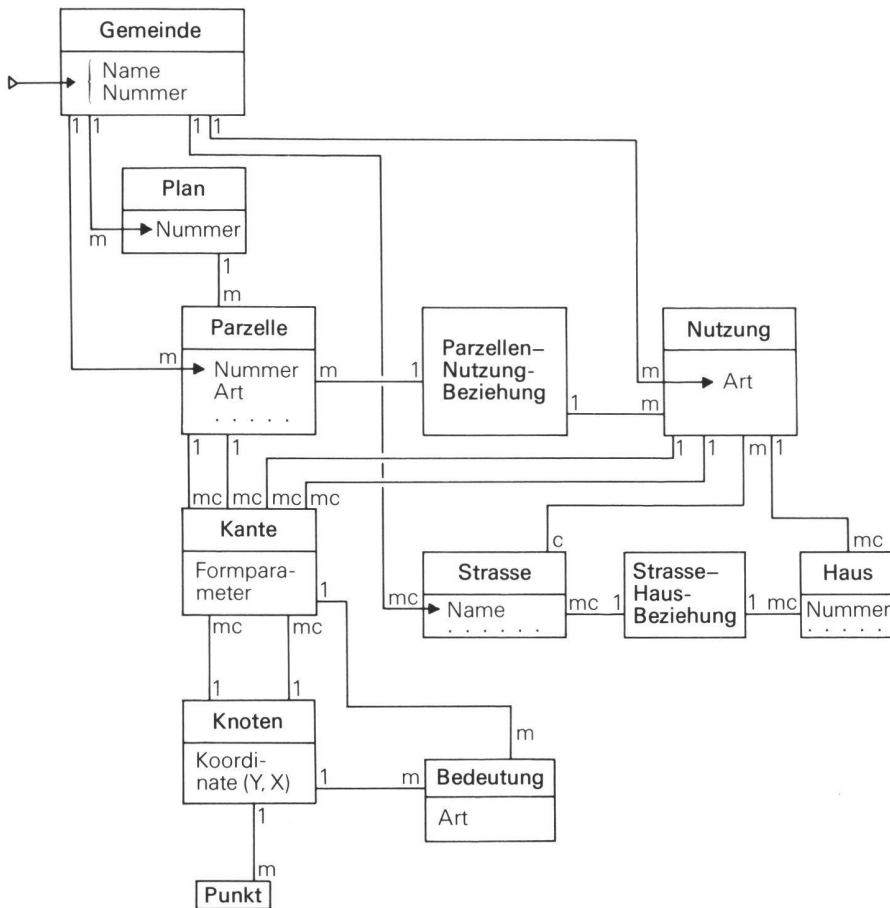


Abb. 22 Erweitertes Entitätenblockdiagramm

Beispiel: Der Benutzer der Datenstruktur in Abb. 22 möchte die (gespeicherten) Daten seines Hauses sehen. Das System liefert ihm: (Parzellennummer), (Strassenname) und (Hausnummer) sowie (Haus-Grundfläche).

### 6.1.3 Physische Ebene

Werden Datensätze auf peripheren Hardwareeinheiten abgespeichert, muss der Computer durch Einführen

von Redundanz sicherstellen, dass er später fehlerhaft gespeicherte Informationen erkennen und korrigieren kann. Schnelleres Auffinden der Datensätze erfordert auf dieser Ebene auch mehr Redundanz und damit Speicherplatz.

### 6.2 Darstellung des Zugriffs zu den Daten

Die Darstellung des Zugriffspfades (wie und in welcher Reihenfolge werden die

einzelnen Datensätze angesprochen) zeigt, auf welche Datensätze (physische Implementierung der Entitäten) der Zugriff optimal ausgerichtet ist und wie Datensätze verknüpft werden können.

### 6.3 Logik

Die gewählte Darstellungsform beschreibt die Daten und ihre Beziehungen einfach und klar. Werden die in 4.3 formulierten Vorschriften eingehalten, so sind bereits wichtige Konsistenzbedingungen erfüllt, die an die Daten gestellt werden müssen.

### 7. Resolution

Die Automationskommission vertritt die Auffassung, dass die vorgestellten Empfehlungen zur Darstellung logischer Datenstrukturen bei EDV-Programmen, die allenfalls bewilligungspflichtig würden, vorzuschreiben seien. Sie unterbreitet diese Darstellungsvorschriften der Projektleitung RAV und ihren Experten sowie der Interessengemeinschaft EDV mit der Bitte, sich intensiv damit auseinanderzusetzen und dazu Stellung zu nehmen.

### Literatur

[Bauknecht 83]: Bauknecht, K./Zehnder, C.A.: Grundzüge der Datenverarbeitung, B.G. Teubner, Stuttgart 1983

[Zehnder 83]: Zehnder, C.A.: Informationssysteme und Datenbanken, Verlag der Fachvereine, Zürich 1983

[Frank 82]: Frank, A.U.: Datenstrukturen für Landinformationssysteme – Semantische, topologische und räumliche Beziehungen in Daten der Geo-Wissenschaften, Diss. ETH Nr. 7197, 1982.

Adresse des Verfassers:

Dipl. Ing. B. Sievers

Institut für Geodäsie und Photogrammetrie  
ETH-Hönggerberg, CH-8093 Zürich

# Recommandations de la commission d'automatisation de la SSMAF à propos de la représentation des structures logiques de données

B. Sievers\*

Au début de ces recommandations, la problématique, qui est à maîtriser lors de l'analyse de la structure logique des données, est mentionnée. Ensuite d'importantes notions sont définies et leur représentation graphique expliquée. En même temps les règles, qui appuient la construction logique de la structure des données, sont énumérés. Un exemple illustre la représentation du diagramme structuré des entités. En conclusion les recommandations traitent les avantages des formes de représentations présentées.

\*B. Sievers a rédigé ces recommandations en se basant sur les délibérations de la CA SSMAF. Les bases de travail étaient données par [Zehnder 83] et [Frank 82]. La traduction française est dû à l'initiative de Messieurs A. Bercher, R. Durussel et B. Studemann. Qu'ils en soient remerciés.

## 1. Introduction, objet

Toute application de l'informatique s'appuie sur un modèle de la réalité adapté au but poursuivi. On définit dans ce modèle quelles sortes de données décrivent les objets et sous quelle forme.

Concernant les développements et la documentation, un postulat de plus en plus important exige une indépendance entre données et installations. On parle ainsi de structure logique des données en opposition à leur organisation physique dans l'ordinateur.

Pour comprendre un système informatique, il est important de connaître sa structure de données. C'est pourquoi une représentation unifiée de ces structures est de première importance. Une telle représentation profite également à la logique du système. Elle entraîne la mise en évidence de relations entre les données.

Tant que le traitement informatique consistait dans le fonctionnement séquentiel de programmes isolés et que ces programmes traitaient des structures simples, la représentation de la structure des données n'était pas essentielle. Les systèmes actuels et futurs maîtriseront des masses de données toujours plus grandes et complexes. La représentation claire de leurs structures devient, donc, de première importance.

## 2. But de ces recommandations

Une représentation unifiée des structures logiques de données doit montrer et rendre compréhensible le modèle qui est à la base d'un système de données. Plus concrètement, il faut clarifier les concepts y relatifs. Le but de ces recommandations n'est cependant pas – et nous insistons sur ce point – de recommander ou de rejeter les modèles cités en exemple. C'est un devoir auquel s'attacheront d'autres institutions.

## 3. Délimitation

Comme mentionné, il s'agit de décrire la logique des systèmes de données. Il convient donc en première ligne de donner des informations sur le système; *les recommandations se concentrent avant tout sur la forme de la représentation.*

Si, plus tard, des règles devaient quand même être élaborées, ces recommandations serviront à améliorer leur constitution logique. Il faut en particulier souligner qu'aussitôt que les structures logiques de données sont destinées à la mise en place de banques de données modernes, il existe des prescriptions formelles impératives.

Si l'on ne tient pas compte de cela, on renonce aux conditions de consistance des données ce qui est contraire au

concept d'une banque de données et qui peut affecter la mise à jour de celles-ci.

## 4. Les concepts et leur représentation

### 4.1 Entité et types d'entités

Une entité est un élément individuel du monde réel ou abstrait. Dès qu'une relation entre entités a un sens dans le monde réel, une de ces relations peut être considérée elle-même comme une entité [Zehnder 83, p. 30].

La notion d'entité est le concept usuel pour décrire dans le cadre d'un modèle une part apparemment définie du monde réel. Dit plus simplement, une entité est une chose bien définie appartenant à un modèle réel. Elle est en général décrite par divers attributs dont la valeur est connue: par exemple le bâtiment, caractérisé par l'adresse «Rue de la Gare 21».

Des entités avec des attributs égaux ou semblables, mais de valeurs diverses, peuvent être assemblées en *types d'entités*.

Formons, par exemple, le type d'entités **Bâtiment** dans lequel nous pouvons différencier les divers bâtiments par leurs attributs individuels.

Les types d'entités sont représentés par des petites cases, dans lesquelles figure le nom du type (en majuscules ou imprimé en gras) (Fig. 1).

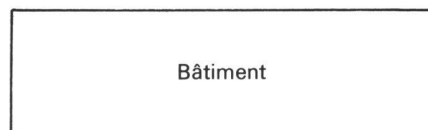


Fig. 1 Représentation d'un type d'entités

Dès que cela est indispensable, on figure les attributs caractéristiques, comme par exemple, les données administratives concernant un bâtiment (Fig. 2).

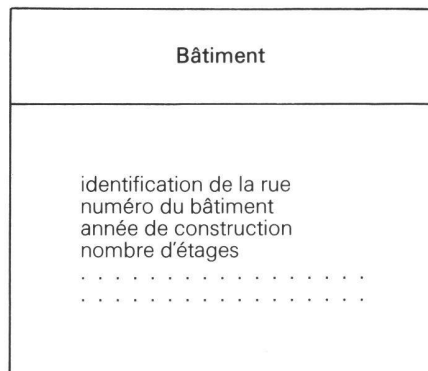


Fig. 2 Représentation d'un type d'entités et de ses attributs

## 4.2 Relations

La description de la structure des données ne consiste pas uniquement dans l'énumération des parties, donc des divers types d'entités, mais aussi dans la description des relations entre eux.

Si des relations existent entre deux types d'entités (comme par exemple commune-canton), elles sont représentées par des traits entre les cases concernées (Fig. 3).

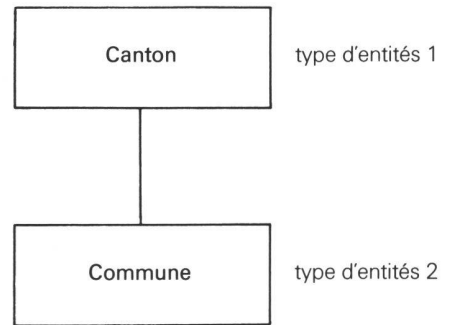


Fig. 3 Représentation d'une relation entre deux types d'entités

On classe les relations en testant combien d'entités d'une catégorie peuvent se relier avec une entité d'un autre type. On en déduit le type d'association. On différencie quatre types d'association pour la relation entre un type d'entités 1 et un type d'entité 2.

- 1: association simple  
une et une seule entité du type 1 est exactement associée à une et une seule autre entité du type 2.
- c: association conditionnelle  
une entité du type 1 est associée à une ou aucune entité du type 2.
- m: association multiple  
une entité du type 1 est associée à plusieurs (mais au minimum à une) entités du type 2.
- mc: association conditionnelle-multiple  
une entité du type 1 est associée à plusieurs, à une ou, également, à aucune entités du type 2.

De telles associations sont aussi bien possibles du type d'entités 1 vers le type d'entités 2 qu'inversement. Pour chaque association entre deux types on définit donc une paire de types d'association, qui partent une fois de l'une et une fois de l'autre entité.

Exemple: A chaque canton appartient une ou plusieurs communes: type d'association m. Dans l'autre sens à chaque commune correspond exactement un canton: type d'association 1. La relation canton-commune est donc au total de type 1: m.

Cela est représenté comme suit dans le diagramme structuré des entités (Fig. 4).

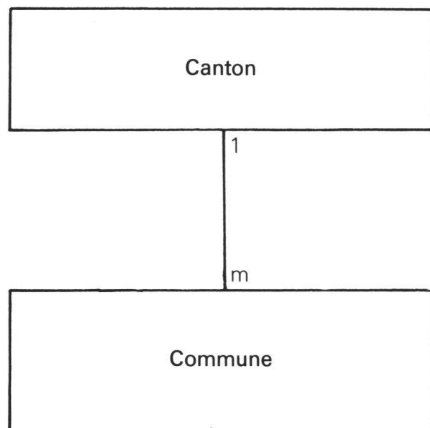


Fig. 4 Représentation des types d'association d'une relation entre deux types d'entités

Dans des cas exceptionnels, on pourra indiquer, à la place d'une valeur indéterminée *m*, une valeur déterminée ou une série de valeurs.

Entre deux entités plusieurs relations sont possibles: comme, par exemple, pour des objets, dont la surface est décrite par un type d'entités: chaque entité **Surface** est séparée d'une entité **Surface** voisine par une entité **Ligne**. La relation de voisinage peut donc être représentée comme suit (Fig. 5).

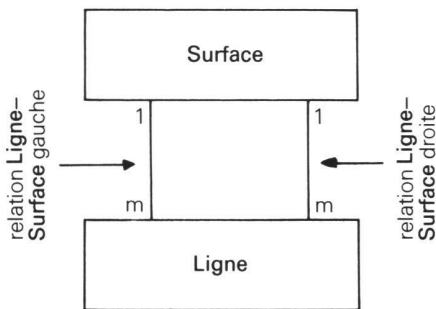


Fig. 5 Représentation des relations entre les types d'entités **Surface** et **Ligne**

Le modèle décrit à la figure 5 exige que les surfaces restantes situées à l'extérieur du modèle soient mémorisées. Si ce n'est pas le cas, la relation **Surface-Ligne** doit être du type 1:mc: en bord de modèle, les lignes ne délimitent alors qu'une seule surface.

### 4.3 Les attributs

Un attribut décrit une caractéristique déterminée d'un certain type d'entités. Exemple: l'entité **Bâtiment** reçoit par les attributs «nombre d'étages», «nombre d'appartements», «année de construction», etc. ... les caractéristiques inhérentes au modèle choisi (Fig. 6).

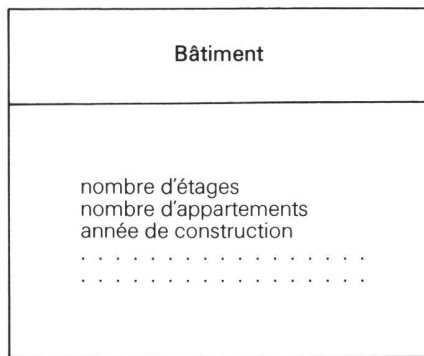


Fig. 6 Représentation d'un type d'entités et de ses attributs

Si un spécialiste en mensurations se penche sur un problème de structures de données pour un système informatique qui devra gérer et mettre à jour des masses importantes de données, il devrait respecter les règles suivantes lors de la détermination des attributs:

4.3.1 Un attribut ne doit avoir qu'une valeur. Si ce n'est pas le cas, il doit être divisé en plusieurs attributs.

Dans l'exemple de la figure 6, l'attribut «année de construction», d'une maison construite en 1920 et transformée en 1960, ne peut avoir simultanément les valeurs «1920» et «1960». Cela devient seulement possible si l'on divise «année de construction» en «année d'édification» et «année de rénovation».

4.3.2 La valeur des attributs doit être choisie parmi des constantes appartenant à une série de valeurs fixées. Si une valeur est attribuée à un attribut, elle ne pourra être remplacée que par une nouvelle constante appartenant à la même série de valeurs.

Dans l'exemple de la figure 6, l'attribut «nombre d'étage» peut se voir attribuer une valeur de la série «nombres positifs entiers entre 1 et 100».

4.3.3 Une série de valeurs pour attributs doit être stable: lors de la définition des données, on fixera une population de valeurs appropriées ne se modifiant pas au cours du temps.

Exemple: la série de valeurs de la coordonnée X d'un point situé en Suisse est 62 000.0000 ... 302 000.0000. On fixe en même temps l'unité (par exemple, le mètre avec quatre chiffres après la virgule).

4.3.4 Un attribut ne doit pas présenter des valeurs qui constituent elles-mêmes des types d'entités: quand une caractéristique est liée par un nombre indéterminé à une entité, elle doit constituer une entité en elle-même. Exemple (Fig. 7)

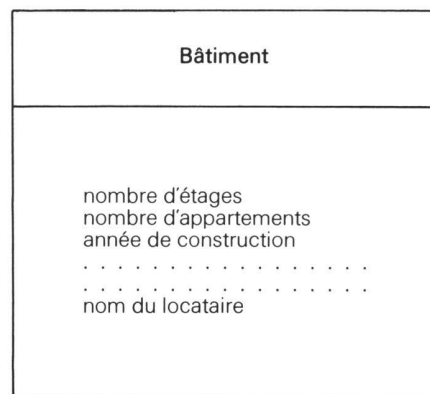


Fig. 7

La figure décrit un modèle, dans lequel chaque bâtiment ne peut avoir qu'un seul locataire. Dans le modèle de la figure 8, cette restriction n'existe plus (Fig. 8).

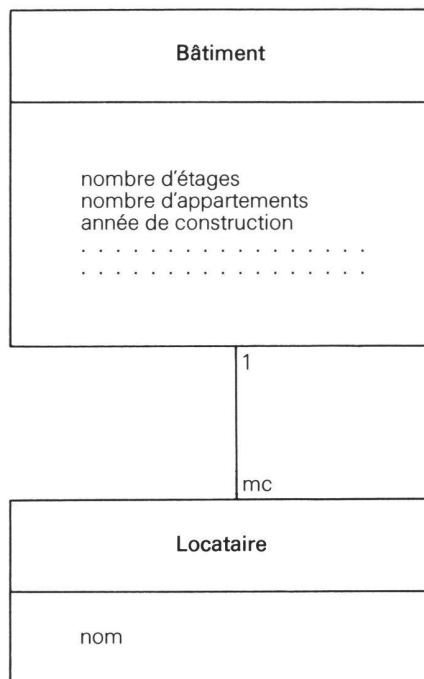


Fig. 8

4.3.5 L'attribut doit être dépendant de l'entité: si l'entité est fixée, la valeur de l'attribut doit l'être par conséquent.

Les attributs ou les combinaisons d'attributs ne doivent pas être fonctions les uns des autres, que ce soit pour le même ou pour divers entités:

Une *dépendance complète* est réalisée, lorsqu'un attribut ou une combinaison d'attributs est fonction d'un autre attribut ou combinaison d'attributs du même type d'entités.

Exemple: le type d'entités **Personne** est identifié par les attributs «nom», «prénom» et «année de naissance». Si l'on introduit encore le groupe d'attributs «numéro AVS», sur lequel chaque groupe de chiffres est introduit de manière

individualisée (code du nom, année de naissance, code de sexe et de naissance, etc.), on constate une dépendance complète entre les dits attributs.

Une dépendance externe est réalisée lorsque A et B décrivent divers types d'entités, mais sont liés entre eux.

Exemple: l'adresse d'une **Personne** ne doit pas contenir à la fois le «nom de la localité» et le «numéro postal» car le «numéro postal» dépend visiblement du «nom de la localité» et est, par exemple, défini dans le type d'entités «*cercle de distribution postal*» (quartier, localité, fraction, ...)

(Le chapitre 6.1 s'occupe des redondances utiles et justifiées.)

4.3.6 Un attribut déterminé ne doit apparaître que dans un seul type d'entités. S'il intervient dans deux ou plusieurs types, comme par exemple le nom de rue dans **Bâtiment** et **Rue** (figure 9):

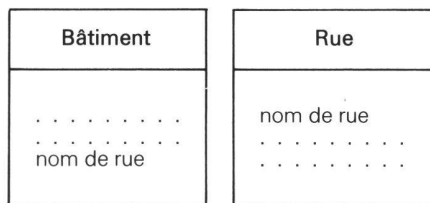


Fig. 9

ceci indique des redondances et des dépendances entre ces types qui sont mal appropriés. Nous pouvons éviter ceci en modifiant l'exemple comme suit (figure 10).

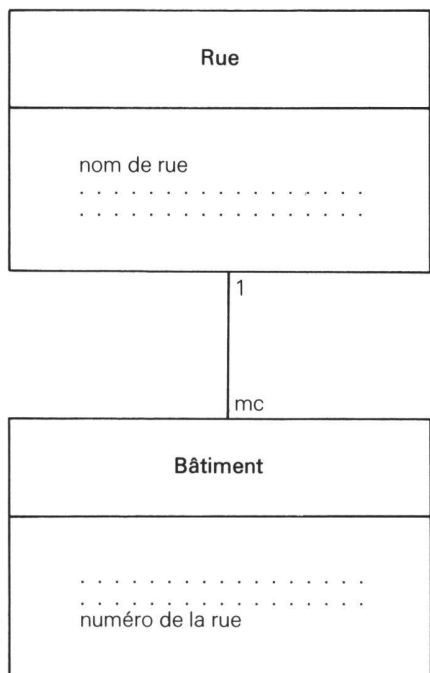


Fig. 10

Par cette relation, le nom de rue appartenant à chaque **Bâtiment** peut être en tout temps déterminé. Il ne doit dès lors plus apparaître comme attribut dans l'entité **Bâtiment**. (Le modèle qui sert de base à la figure 10 stipule que chaque bâtiment n'est situé qu'au bord d'une seule rue.)

#### 4.4 Représentation de l'accès aux données

Le diagramme structuré des entités doit également laisser apparaître comment l'on peut accéder aux données ainsi structurées.

Si l'on cherche une entité déterminée, deux possibilités se présentent:

- partir du type d'entités supérieur et chercher le type d'entités qui lui est subordonné en se basant sur l'entité recherchée. (Exemple: on cherche le numéro de téléphone de «Jean Dupont» en ville de Genève: le type supérieur est le registre des abonnés de la Suisse, le type subordonné est constitué de tous les abonnés du bottin de téléphone de Genève.)
- on utilise des attributs ou des combinaisons d'attributs fixés de manière unique pour mettre en place l'organisation de la mémorisation des données (par exemple l'organisation «Hash» dans laquelle l'adresse de la mémoire d'une entité est une fonction d'un de ses attributs caractéristiques). Ce mode d'organisation épaulé efficacement une recherche rapide. Dans un diagramme structuré des entités, on représente cet appui à l'accès comme suit:

pour un attribut caractéristique unique (Fig. 11).

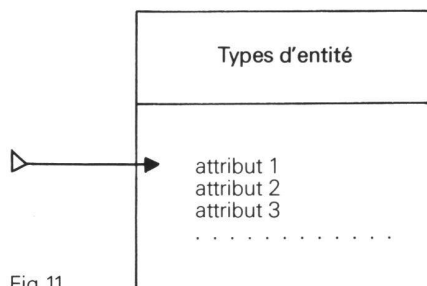


Fig. 11

pour une combinaison caractéristique unique d'attributs (Fig. 12).

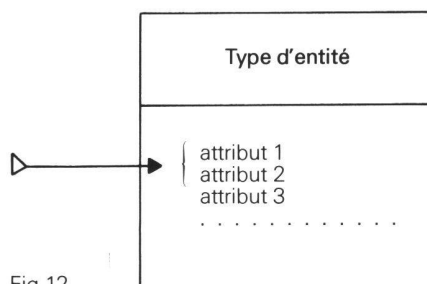


Fig. 12

Si, par contre, les attributs ou les combinaisons d'attributs ne sont pas uniques pour l'ensemble du modèle, mais seulement pour un domaine particulier, on peut le représenter aussi comme suit (Fig. 13).

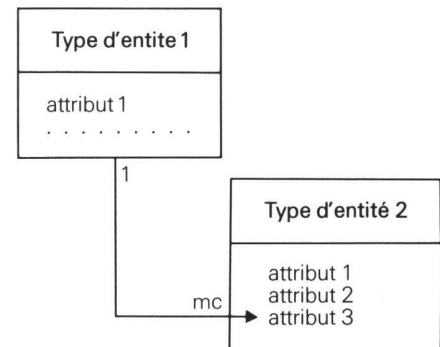


Fig. 13

Exemple: le nom de rue n'est, dans la règle, caractéristique qu'à l'intérieur d'une commune (Fig. 14).

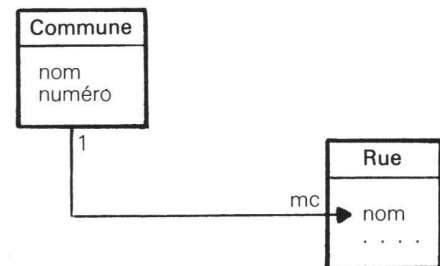


Fig. 14

#### 4.5 Généralisation

La forme de représentation de structures de données compliquées peut, selon [Frank 82], être rendue plus lisible par la généralisation: elle se réalise graphiquement par l'imbrication des cases contenant les entités, dans laquelle les types d'entités imbriqués possèdent tous les attributs des types d'entités qui les dominent hiérarchiquement (Fig. 15).

**Point de polygone** a donc les attributs «numéro», «coordonnées (Y, X)», «altitude» et genre de «matérialisation» et éventuellement le type subordonné **Coordonnées d'appoint**.

#### 4.6 Diagramme structuré des entités

La représentation d'une structure logique de données avec les cases des types d'entités, les lignes indiquant les relations et la description des types d'association est appelée *diagramme structuré des entités*. Le *diagramme structuré élargi des entités* contient en plus les accès aux données et la généralisation graphique.

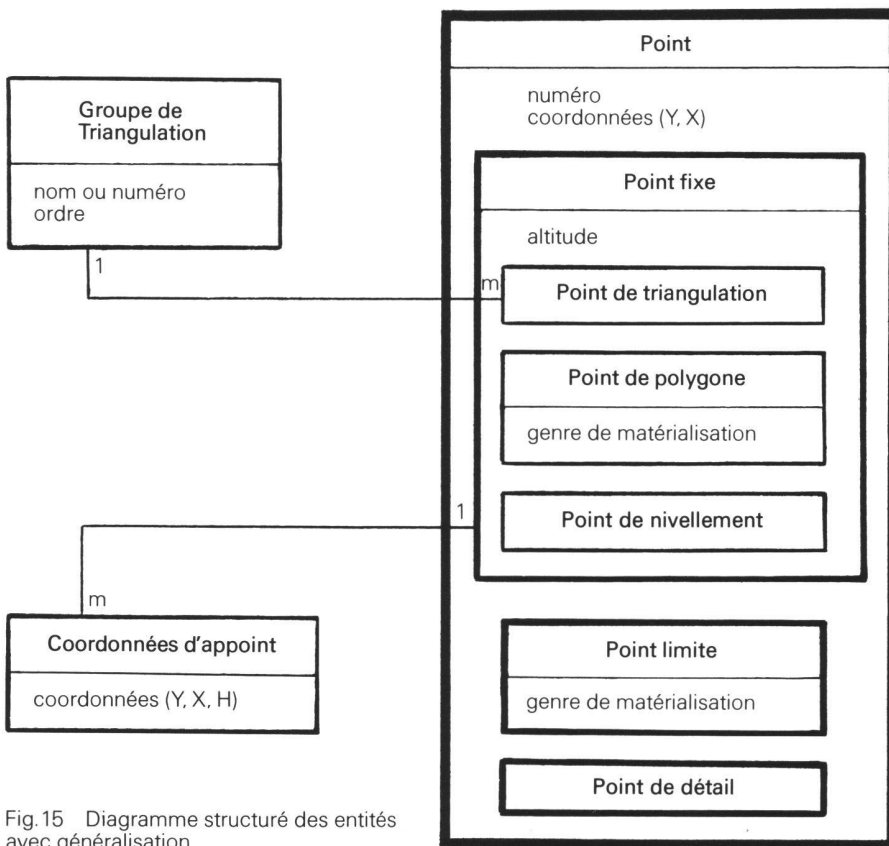


Fig. 15 Diagramme structuré des entités avec généralisation

seuls points contenus dans le modèle seraient des points limite et des points de détail (Fig. 16).

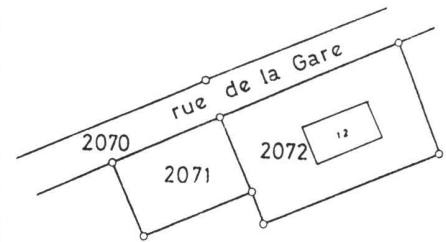


Fig. 16 Exemple d'un modèle du monde réel

D'ailleurs voici un premier projet du diagramme structuré des entités (Fig. 17).

Essayons de sonder les possibilités de l'exemple de la fig. 17:

Pour un bâtiment ou une rue constituant également une parcelle, les informations géométriques (coordonnées des points, définition des lignes) doivent être disponibles de façon séparée pour les trois notions.

#### 4.7 Redondance

«Dans un ensemble de données, on parle de redondance quand une partie des données peut être écartée sans perte d'information» [Zehnder 83, p. 38]. Des données redondantes sont donc des données qui sont mémorisées plusieurs fois, ou qui peuvent être déduites d'autres données déjà mémorisées. On constate également des redondances lorsque les règles 4.3.5 et 4.3.6 ne sont pas respectées (le paragraphe 6.1 traite des redondances justifiées).

#### 4.8 Consistance

«Nous comprenons par là la concordance du contenu de la base de données avec la description des données et l'absence de contradiction dans les données» [Zehnder 83, p. 132] (comparez à ce propos le 3e alinéa du paragraphe 3).

#### 5. Application à un exemple simple

Supposons que notre modèle du monde réel puisse être décrit au moyen d'un plan. Des parcelles de rues, des bien-fonds, ainsi que des bâtiments, avec leur contour périmétrique, formeraient le contenu du dit modèle. Les

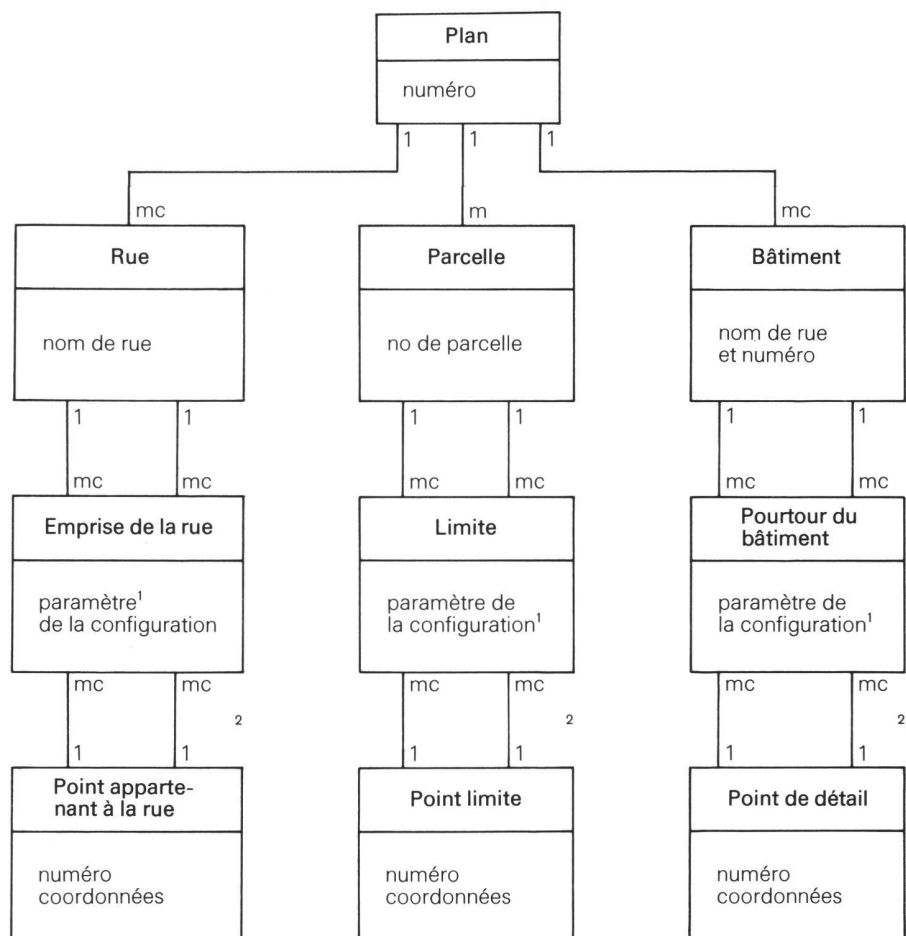


Fig. 17 Exemple d'un diagramme structuré des entités concernant la figure 16

<sup>1</sup> La sinuosité d'un parcours est décrite par le paramètre de la configuration.

<sup>2</sup> (Fig. 18)

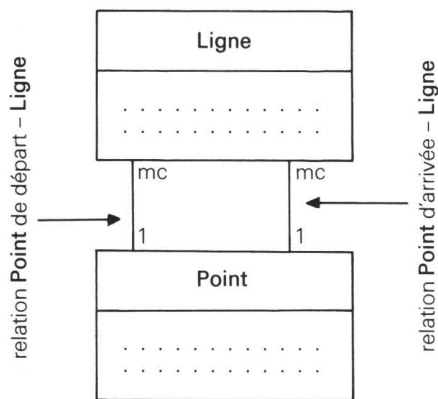


Fig. 18

Certains attributs (nom de rue, paramètre de la configuration, numéro et coordonnées des points) apparaissent dans plusieurs types d'entités. Pour les mutations, la totalité de la base de données doit alors être vérifiée du point de vue de sa consistance.

Des relations de voisinage (p. ex. bâtiment 12 est situé dans la parcelle 2072, parc. 2071 est en limite d'une rue) ne sont pas introduites.

Les bâtiments doivent avoir une adresse simple et précise, chaque bâtiment doit se trouver en limite d'une rue. (Cette condition a été adoptée dans le modèle présenté. D'autres possibilités peuvent être envisagées grâce à l'incorporation du numéro d'assurance ou d'une éventuelle double adresse.)

Pour améliorer ce diagramme structuré des entités, il faut essayer de définir les identités et d'éviter le stockage multiple de mêmes données (Fig. 19).

Avec l'introduction d'un type d'entités **Utilisation du sol**, nous avons créé la possibilité de définir une parcelle à partir des différentes surfaces de nature qui la composent (p. ex. bois, pré, bâtiment, chemin, ...). Ce modèle exige une subordination des surfaces de nature à la parcelle: une surface de nature peut être au maximum aussi grande que celle de la parcelle, ou encore la surface d'une parcelle peut se composer de plusieurs surfaces de natures différentes.

Le passage aux types d'entités **Arête** et **Point** permet de saisir les tronçons de limite et de nature identiques (avec point de départ et d'arrivée identique) en une seule fois.

L'incorporation du type d'entités **Classe des points** permet de distinguer des points de différentes significations (points limite, points de détail). Par contre, si un point physique réunit des points de différentes classes, chaque classe d'un point doit être mémorisée (selon Fig. 19).

La relation **Rue-Utilisation**: une rue se compose de plusieurs surfaces d'utilisation. Une surface d'utilisation ne peut appartenir au maximum qu'à une seule rue, ce qui signifie que les carrefours sont à partager entre les rues.

La relation **Rue-Bâtiment**: la remarque relative à la Fig. 10 est aussi valable ici. Pour les modèles, dans lesquels chaque bâtiment borde une rue, ou un bâtiment borde plusieurs rues, le diagramme structuré des entités suivant peut être utilisé (Fig. 20).

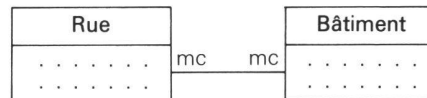


Fig. 20

L'implémentation de relations m/mc-m/mc ou c/mc-c/mc entre des types d'entités présente des difficultés. Le spécialiste de l'informatique a résolu ces relations par l'incorporation d'un type d'entités intermédiaire (Fig. 21).

Dans un dernier exemple, les faiblesses du modèle dépistées jusqu'ici sont éliminées.

Description du modèle:

Dans une commune dont le périmètre est divisé en plans, les parcelles et les surfaces de natures doivent être introduites au même niveau. Les points, les rues et les bâtiments sont à mettre

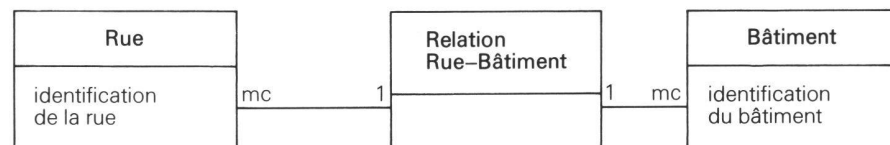


Fig. 21

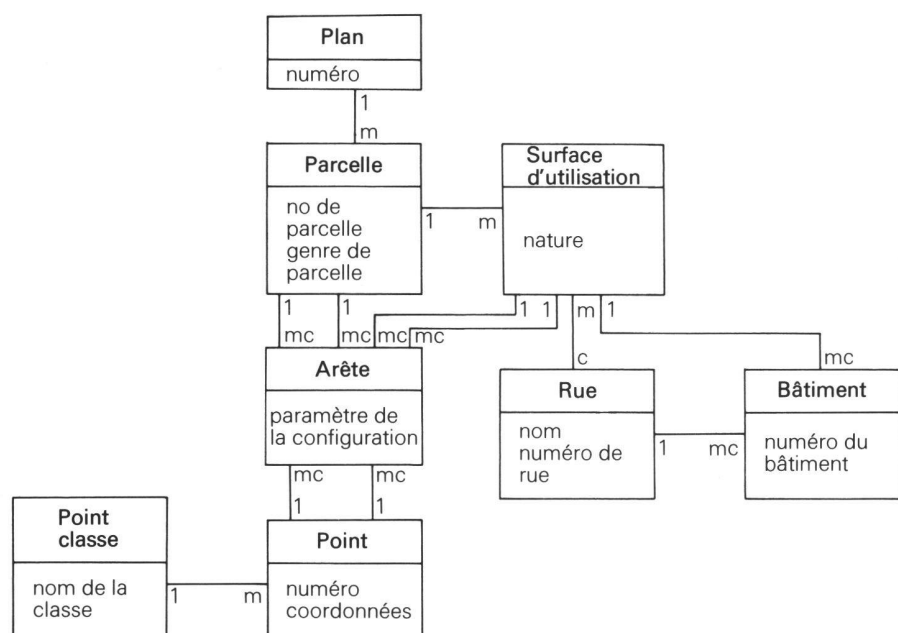


Fig. 19 Exemple amélioré d'un diagramme structuré des entités concernant la figure 16

particulièrement en évidence. Une relation entre les rues et les bâtiments est envisageable; entre parcelles et bâtiments, celle-ci doit être introduite (à l'aide du type d'entités **Utilisation**). L'utilisation doit être indiquée par parcelle, mais devrait aussi pouvoir être introduite indépendamment des parcelles.

Les restrictions restantes, vis-à-vis de ce modèle, sont les suivantes:

- Une parcelle ne peut pas s'étendre sur plusieurs plans\*.
- Les constructions souterraines ne peuvent pas être introduites. Il en est de même pour les divisions à l'intérieur des bâtiments.
- La numérotation des points doit être bi-univoque.

\* Cette restriction peu réaliste dans la pratique pourrait être supprimée d'une façon simple: entre le type d'entités **Plan** et le type d'entités **Parcelle** on ajoute un type d'entités (intermédiaire).

- Les numéros de plans et des parcelles, ainsi que les noms de rues doivent être uniques à l'intérieur d'une commune.

La structure des données de ce modèle est présentée dans la figure 22.

Commentaires: Le type d'entités **Signification** porte l'information sur la (na-

dances inutiles qui subsistent encore au niveau conceptionnelle du système.

On distingue, en analysant la redondance utile, trois niveaux:

- niveau conceptionnelle
- niveau utilisateur (niveau externe)
- niveau physique de la base de données (niveau interne)

des de représentations combinées et étendues, avec une redondance suffisante.

Exemple: L'utilisateur de la base de données (Fig. 22) désire voir les données mémorisées de son bâtiment. Le système lui indique: «numéro de parcelle», «nom de rue», «numéro du bâtiment» ainsi que «surface du bâtiment».

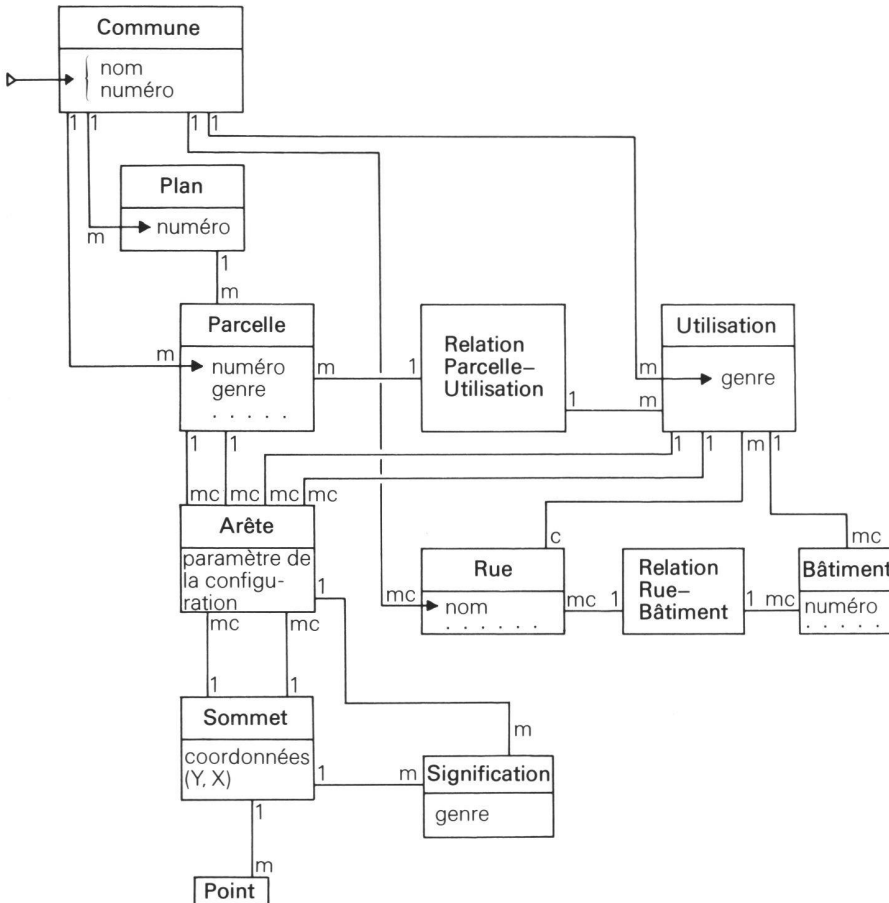


Fig. 22 Diagramme structuré élargi des entités

ture) de l'entité **Arête** et ses points extrêmes (par ex. «rue», «parcelle», «bâtiment»). Le type d'entités **Sommet** permet de détecter immédiatement des points de classes différentes pour un même point physique (par ex. point de polygone = point limite = point de détail).

La représentation complète de la figure 15 est remplacée par la case **Point** où l'attribut «coordonnées (Y, X)» du type d'entités **Point** est déplacé dans le type d'entités **Sommet**.

## 6. Avantages du diagramme structuré des entités

### 6.1 Redondance utile

Une claire représentation formelle de la structure des données permet de mettre en évidence et d'éliminer les redon-

#### 6.1.1 Niveau conceptionnelle

L'analyse et la représentation de la structure logique de données appartient à ce niveau. Le but est de réaliser un système de données consistant. Le danger «d'anomalies de mutation» (des données redondantes ne sont pas mutées ou ils ne peuvent pas être mutées à cause de la redondance) doit être mis en balance avec la sécurité des données (des données non redondantes, après endommagement, ne sont presque jamais récupérables) et avec la rapidité d'accès. Après cette analyse ne subsistent que les redondances utiles.

#### 6.1.2 Niveau utilisateur

L'utilisateur demande d'une base de données qu'elle réponde à ses deman-

#### 6.1.3 Niveau physique

Si les données sont stockées sur une unité périphérique, l'ordinateur doit assurer, par l'introduction de redondances, le dépistage et la correction des informations incorrectes. Un examen plus rapide des données requiert, à ce niveau, encore plus de redondances, donc plus de place mémoire.

## 6.2 Représentation de l'accès aux données

La représentation de l'accès (comment est obtenue une donnée isolée) montre dans quelle série de données (implantation physique du type d'entité) on obtient un accès optimal et comment les données peuvent être combinées.

## 6.3 Logique

La forme de représentation choisie décrit les données et leurs relations de façon simple et claire. Si les instructions formulées au chapitre 4.3 sont observées, d'importantes conditions de consistance, qui doivent s'appliquer aux données, seront remplies.

## 7. Résolution de la commission d'automatisation

L'opinion de la commission d'automatisation est que l'application de ces recommandations pour la représentation des structures des données dans les programmes doit être imposée pour tous les cas de programmes soumis à approbation.

Elle soumet ces directives de représentation à la direction du projet REMO et à ses experts ainsi qu'à CITAIM avec prière d'examiner ensemble ce problème et de prendre position.

#### Littérature

- [Bauknecht 83]: Bauknecht K./Zehnder C. A.: Grundzüge der Datenverarbeitung; B. G. Teubner, Stuttgart 1983.
- [Zehnder 83]: Zehnder C. A.: Informationssysteme und Datenbanken; Verlag der Fachvereine, Zürich 1983.
- [Frank 82]: Frank A. U.: Datenstrukturen für Landinformationssysteme - Semantische, topologische und räumliche Beziehungen in Daten der Geo-Wissenschaften; Diss. ETH Nr. 7197, 1982.

Adresse de l'auteur:

B. Sievers  
Institut de Géodésie et Photogrammétrie  
EPF-Hönggerberg, CH-8093 Zürich