



École interdisciplinaire en numérique de la santé

Des données aux modèles, de la logique aux relations

Une introduction à la modélisation et à la détermination des exigences

EINS_2025

Luc LAVOIE (luc.lavoie@usherbrooke.ca)

—

EINS/Scriptorium/EINS2025-NDC_Lavoie_Luc, version 1.0.0.c, en date du 2025-05-25

— document de travail —

Sommaire

Si la modélisation adéquate des données de santé passe par la logique et les relations, encore faut-il définir les capacités attendues du méta-modèle, puis démontrer l'adéquation de ce méta-modèle en regard des capacités attendues. C'est par cette démarche classique de spécification (exploration, modélisation, clarification, détermination et formalisation) des exigences que nous débuterons l'activité. La deuxième partie nous permettra de caractériser ce méta-modèle et d'en circonscrire les possibilités, les moyens, les défis et les limites. Nous montrerons finalement comment certaines limites pourraient être levées par la modélisation adéquate des connaissances et ainsi préparer le terrain à l'activité suivante.

Les objectifs ciblés sont les suivants :

1. Caractériser le problème de mise en place de systèmes d'information en santé d'un point de vue informatique.
2. Caractériser la solution de prédilection : la modélisation relationnelle.
3. Présenter l'essentiel des possibilités, des moyens, des défis et des limites de la modélisation relationnelle.
4. Motiver le recours aux ontologies appliquées pour la représentation des connaissances soutenant les systèmes d'information en santé.

Mise en garde

Le document est structuré selon deux axes de développement :

- la présentation d'une méthode d'élaboration de modèles de données susceptible de soutenir la spécification et le développement de sous-systèmes informatisés conçus pour le domaine de la santé.
- l'encadrement d'une démarche d'auto-apprentissage à être conduite en atelier tout au long de l'EINS 2025 et qui consiste à expérimenter puis valider (nous en sommes convaincus) l'approche préconisée.

La trame principale du document est celle de la présentation de la méthode, elle sera toutefois annotée à l'aide de balises distinctives :



Une explication ou un point de vue complémentaire.



Une anticipation relative au contenu du document.



Une étape d'auto-apprentissage à réaliser en atelier.

Historique

diffusion	resp.	description
2025-05-25	LL	Édition 2025.
2024-05-30	LL	Édition 2024.
2023-07-11	LL	Ébauche initiale.

Table des matières

Le problème	4
1. Une démarche	5
2. Un cas	6
2.1. Caractérisation du problème	6
2.2. Caractérisation des solutions	8
2.3. Description d'une première solution	12
2.4. Vérification de la solution	20
2.5. Première expérimentation	20
2.6. Conclusions préliminaires	20
3. Trois réflexions	22
Remerciements	23
Références	24

Le problème

- Comment accéder aux données de santé ?
 - Ces données sont-elles suffisantes pour faire la prévention, le soin et la recherche en santé ?
- Comment accéder aux données *requis*es pour la prévention, le soin et la recherche en santé ?
 - La signification d'une donnée est-elle toujours univoque ?
 - Est-ce que la donnée est suffisante en elle-même ?
- Comment accéder aux données *et les interpréter (correctement)* ?
 - Ce problème est-il susceptible d'avoir une solution consensuelle, indépendamment des cultures, des sociétés, des états, des ordres professionnels et des intérêts privés ?
 - Ce problème est-il unique ou ne faudrait-il pas considérer une famille de problèmes, une famille de solutions ?



Constats et pistes

- Plusieurs solutions partielles aux problèmes d'accès aux données coexistent au sein des organisations et des réseaux de santé.
- Quels que soient nos efforts, il est vraisemblable que la couverture de ces solutions demeurera partielle et, pour cette raison, il serait souhaitable qu'elles soient interopérables.
- Quels que soient nos efforts, il est vraisemblable que l'interprétation de ces données demeurera plurielle et, pour cette raison, il serait souhaitable qu'elle se fasse sur la base de processus et de modèles documentés, ouverts et traçables.
- L'usage du conditionnel (« serait souhaitable ») dans les deux derniers constats sous-entend que tel ne serait pas *toujours* le cas présentement. Un euphémisme... en quelque sorte.



Atelier, étape 1

- Puisqu'il y aura plusieurs modèles, il importe de se doter d'une démarche pour les élaborer.
- Puisqu'il y aura plusieurs modèles, il faut déterminer un **méta-modèle** facilitant :
 - le raisonnement,
 - la cohérence,
 - l'interopérabilité,
 - la documentation,
 - l'évolutivité,
 - la traçabilité.



- Acceptez-vous comme fondements
 - La logique des prédicats selon la formulation qu'en a faite Boole ?
 - La théorie des ensembles selon la formulation qu'en a faite Bourbaki ?
 - La théorie des types de Russel reformulée en incluant le sous-typage ?
- Alors, soyez prévenus, le seul méta-modèle connu, satisfaisant aux conditions énoncées à l'étape 1 est la théorie relationnelle.
- De plus, ce méta-modèle se construit et se représente en termes de lui-même au même titre que tous les modèles qu'il permet de définir.

1. Une démarche

1. Caractériser le problème.
2. Caractériser les solutions.
3. Concevoir, décrire et vérifier les solutions.
4. Choisir certaines solutions.
5. Expérimenter et valider celles-ci.
6. Tirer les conclusions.



Atelier, étape 2

Explicitons maintenant notre démarche tout en l'appliquant à notre premier problème :

- Trouver **LE** méta-modèle !

2. Un cas

2.1. Caractérisation du problème

- Cerner un problème afin d'en faciliter la solution passe par l'analyse et la modélisation dudit problème et donc d'une caractérisation initiale susceptible d'orienter l'analyse.
- Comparer un problème à d'autres est riche en enseignement, voire en solutions existantes ou adaptables.
- Choisir une solution, parmi d'autres, passe par la comparaison de celles-ci aux autres solutions.



Deux formes de caractérisation sont prévalentes lors de l'analyse des problèmes et solutions en modélisation de données :

- les dimensions
- les besoins

2.1.1. Les dimensions, caractérisation primaire et partielle

Volume

quantité de données devant être reçues, traitées, stockées, produites.

Variété

diversité et complexité des types utilisés par le modèle.

Vélocité

débit de données entrantes et sortantes.

Variabilité

relativité en termes d'agent, de spatio-temporalité, de perspective, de marge d'erreur et d'incertitude.

Valeur

richesse analytique des données (de laquelle découlent les valeurs personnelles, sociales, scientifiques, politiques, financières, etc.).

Virtualité

agrégation des sources de données distribuées.

Vertu

gouvernance et sécurité des données.



- La mesure de ces dimensions n'est pas aussi triviale qu'on le voudrait.
- Les procédés et les techniques existent, mais ne font pas (encore) consensus.

Dimensions typiques dans le domaine de la santé

Comment ces dimensions se comparent-elles relativement à celles d'autres domaines, tels que la logistique, les télécommunications, le contrôle-commande de systèmes embarqués mobiles ou la production d'énergie ?

- **Volume** : pas considérable *hors génomique et imagerie*
- **Variété** : considérable
- **Vélocité** : pas considérable *hors génomique et imagerie*
- **Variabilité** : considérable
- **Valeur** : *perception très variable*
- **Virtualité** : considérable

- **Vertu**: complexe, sensible, débattue

2.1.2. Les besoins, caractérisation secondaire et partielle



1. Les concepts *objet*, *caractéristique*, *action* et *méthode* sont considérés comme prédéfinis.
2. On distingue trois catégories d'*objets*:
 - objet matériel,
 - objet informationnel,
 - objet procédural.

- **procédé**: *méthode* (normalement systématique, voire formalisée) destinée à produire un ou plusieurs artefacts (ou, plus familièrement, à obtenir un résultat).
- **processus**: ensemble d'activités conforme à un procédé qui en détermine, entre autres, les dépendances.
- **activité**: suite de tâches.
- **tâche**: *action* portant sur un *objet*.
- **effet**: modification d'une ou plusieurs *caractéristiques* d'un *objet*.
- **artéfact**: *objet* ou effet obtenu par l'exécution d'un processus (d'une activité, d'une tâche).
- **besoin**: artéfact considéré nécessaire à la définition d'un procédé ou à l'exécution d'un processus (d'une activité, d'une tâche); *ne pas confondre avec désir, attente ou exigence*.



Noter la récursion, un besoin est un artéfact produit par un processus au bénéfice d'un processus (souvent autre... mais pas forcément) ou du procédé le définissant. En effet, rien n'empêche un processus de produire un artéfact pour lui-même, notamment si celui-ci est itératif ou récursif.

- Les attentes des parties prenantes sont documentées dans le document de vision (DDV) antérieurement à l'analyse des besoins.
- Les exigences sont établies dans la spécification des exigences (SES) postérieurement à l'analyse des besoins.
- Les désirs ne sont généralement pas considérés dans une approche rationnelle, du moins ne devraient-ils pas l'être.
 - Désirer: éprouver la passion [le fantasme] de posséder, de voir se réaliser ou de jouir de quelque chose.
- Voir également [Bray2002a, Hull2011a, Khnaisser2015a, Khnaisser2015b].



Atelier, étape 3

- Les besoins suivants sont souvent mentionnés:
 - soutenir une démarche systématique de modélisation
 - soutenir le raisonnement
 - soutenir l'interopérabilité
 - soutenir la documentation
 - soutenir l'ouverture
 - soutenir la traçabilité, voire l'explicabilité

- Les décrire et les compléter.

2.2. Caractérisation des solutions

- En présence d'une proposition de solution, il faut s'assurer qu'elle en est effectivement une.
- En présence de plusieurs solutions, il faut pouvoir les comparer.



Commençons par une forme de caractérisation, les exigences, et une façon de les utiliser, l'adéquation.

2.2.1. Exigences

Les exigences ont pour but de déterminer :

- les conditions **nécessaires et suffisantes** pour qu'une solution soit acceptable ;
- les caractéristiques **mesurables pertinentes** permettant de comparer deux solutions.

En fait, la mesurabilité est-elle suffisante ?

La catégorisation ne pourrait-elle être utile, voire nécessaire ?

L'intérêt de la catégorisation réside en sa capacité de permettre une analyse (et une comparaison) qualitative.

Procédés

Plusieurs procédés ont été proposés afin de permettre la détermination des exigences. Ces procédés se distinguent par l'organisation des activités suivantes et les méthodes préconisées pour chacune.

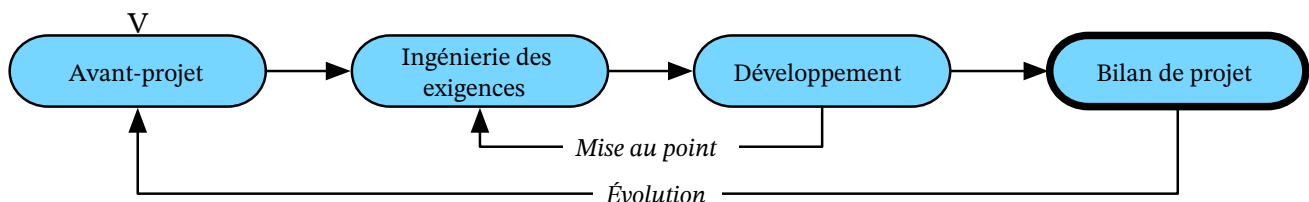
1. exploration
2. analyse
3. spécification
4. vérification
5. validation

Cycle de vie

Ces procédés spécifiques s'insèrent dans un contexte plus large qui embrasse la totalité du cycle de vie du produit (logiciel, sous-système, voire système).

Les quatre prochains diagrammes illustrent ce contexte.

2.2.1.1. Vue globale

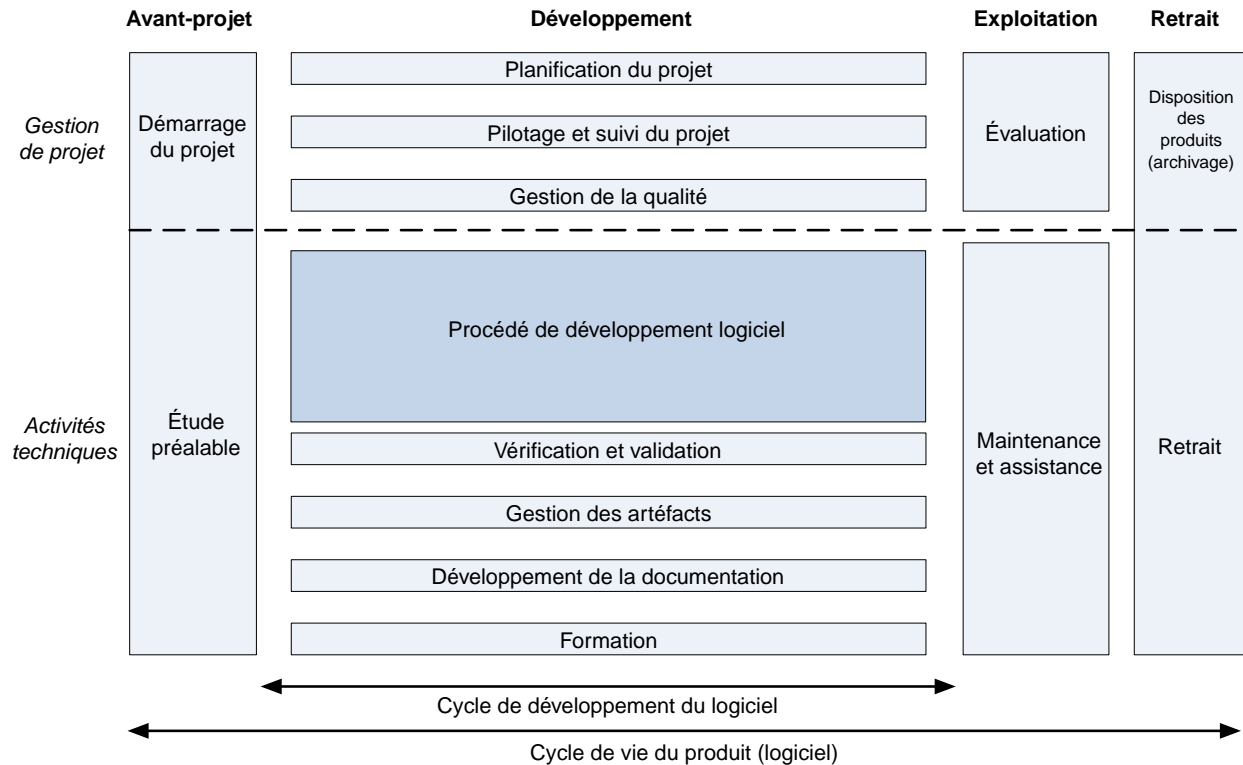


Prendre note que la vue globale précédente omet plusieurs aspects du cycle de vie, dont la formation, la livraison, le déploiement, le soutien, l'exploitation, le retrait, etc.

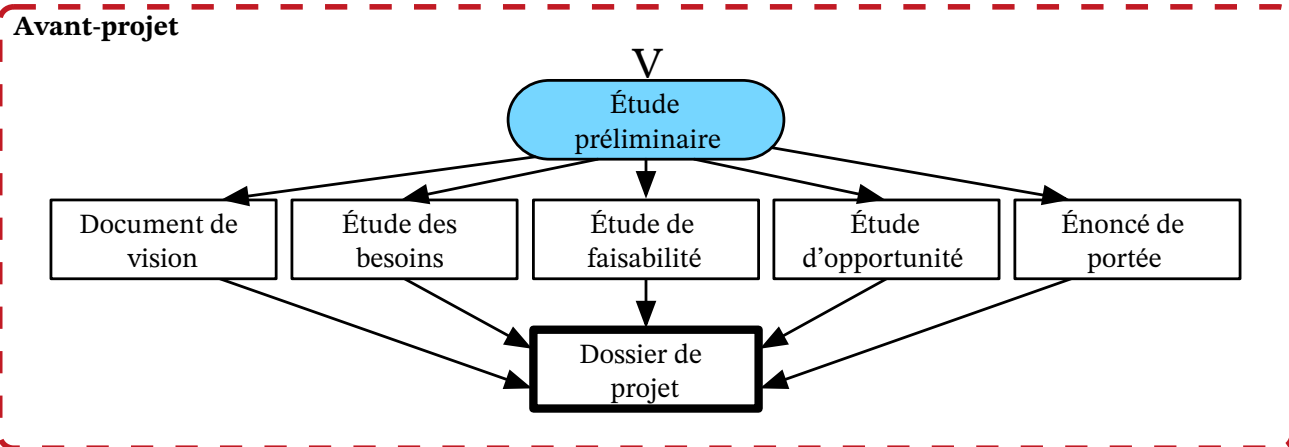
Le diagramme suivant, beaucoup plus complet, a été proposé en son temps par l'IEEE.

CVL_IEEE.graffle [Base] (2025-01-13)

Cycle de vie du logiciel

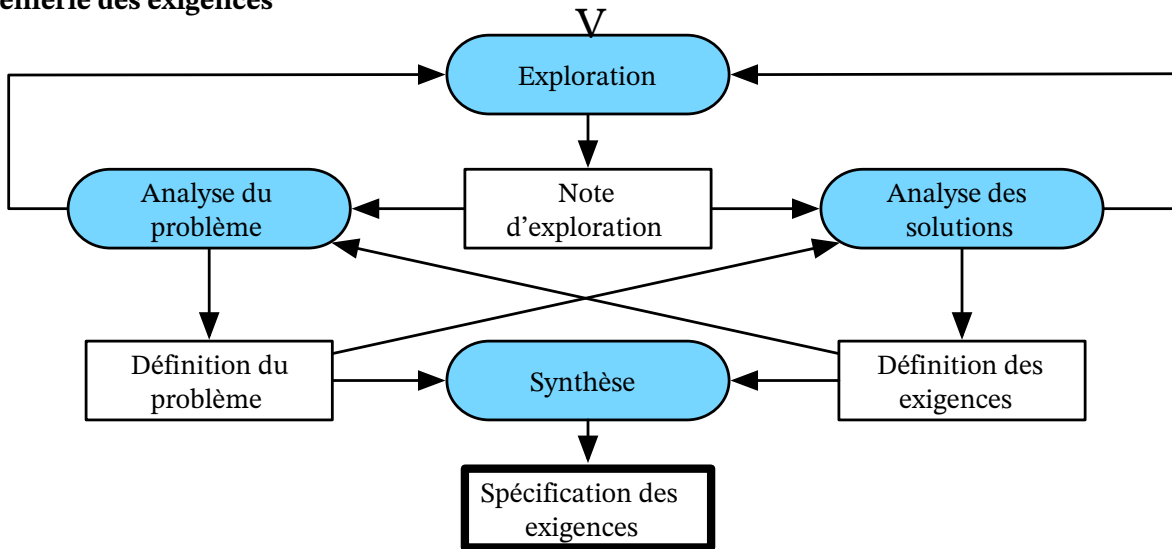


2.2.1.2. Avant-projet



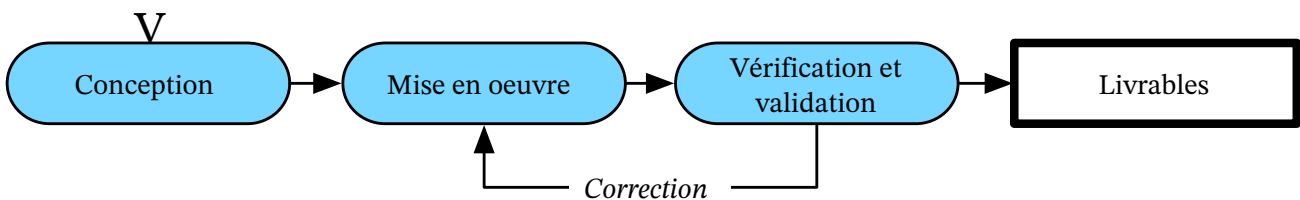
2.2.1.3. Ingénierie des exigences

Ingénierie des exigences



2.2.1.4. Développement

Développement (*)



(*) Un grand nombre de livrables sont produits par les différents processus. Ils varient considérablement selon le procédé utilisé et la solution retenue.

Atelier, étape 4

Exigences pratiques



- Capacité de soutenir un processus de diagnostic puis de choix de traitement (fondée sur des données probantes en regard de pratiques médicales reconnues).
 - Corolaire 1 : capacité de décrire l'état du patient.
 - Corolaire 2 : capacité de formuler un diagnostic.
 - Corolaire 3 : capacité de décrire un traitement.

Atelier, étape 5

Exigences induites A



- Capacité de transposer les théories et les modèles scientifiques dans un cadre unifié (appelé modèle).
 - Corolaire 4 : le méta-modèle doit fournir un formalisme permettant de décrire les théories et les modèles scientifiques.

- Capacité de comparer deux modèles.
- Capacité d'évaluer l'adéquation d'un modèle.

Atelier, étape 6

Exigences induites B



- Capacité de formuler des hypothèses et de les vérifier (au sein du modèle).
 - Corolaire 5: le méta-modèle doit permettre le raisonnement et l'utilisation de données probantes en regard de théories et de modèles scientifiques décrits à l'aide du formalisme.
 - Considération pratique: il serait souhaitable que le méta-modèle permette la vérification automatisée de prédicats en regard de données.

Remarquons que les quatre capacités induites sont applicables tant au méta-modèle qu'aux modèles qu'il permet de décrire (de modéliser).



L'exploration n'est pas terminée :

- une clarification est certainement possible ;
- d'autres exigences découlent des besoins exposés.

La spécification doit s'ensuivre afin

- de s'assurer que chaque exigence soit *claire, exacte, complète et concise* ;
- d'associer à chaque exigence un critère objectif permettant de vérifier qu'elle est satisfaite.

La vérification et la validation doivent s'ensuivre afin

- de minimiser les risques de devoir recommencer la modélisation qui en découlera.

Explicitons :

- claire (c'est-à-dire lisible et compréhensible),
- exacte (c'est-à-dire précise et sans erreurs),
- complète (c'est-à-dire comprenant tous les éléments requis et tous les éléments nécessaires) et
- concise (c'est-à-dire sans éléments superflus).

2.2.2. Adéquation

L'adéquation détermine à la fois la façon d'utiliser les exigences, de les concilier, mais aussi de les compléter au moment d'une évaluation globale d'une solution.

Critères absolus

1. Validité
 - conformité au modèle.
2. Efficacité
 - conformité aux exigences.
3. Cohérence

- non-contradiction interne.

Critères relatifs

4. Complétude
 - couverture « suffisante » du problème.
5. Efficience
 - consommation « acceptable » de ressources.
6. Évolutivité
 - adaptation « aisée » aux changements.

Méta-critères

7. Réfutabilité
 - comporte les mécanismes permettant d'exprimer et d'évaluer les « falsificateurs potentiels ».
8. Acceptabilité
 - comporte les mécanismes permettant d'exprimer et d'évaluer les « critères éthiques ».

Synthèse

- Les critères relatifs sont souvent décomposés en sous-critères auxquels sont associés un protocole de mesure et un poids de façon à permettre une évaluation globale pondérée. D'autres méthodes d'évaluation globale sont possibles.
- La réfutabilité est considérée en regard d'une épistémologie donnée; en conséquence, une solution doit préciser son cadre épistémologique.
- L'acceptabilité est considérée en regard d'une éthique donnée; en conséquence, une solution doit préciser le cadre éthique.

- Examiner la pertinence de compléter avec des catégorisations permettant de faire une analyse et une comparaison qualitative.

Atelier, étape 7

Aux fins de l'exercice, il est suggéré d'utiliser :

- Pour l'épistémologie : celle de Karl Popper, avec la réfutabilité discrète (de préférence binaire).
- Pour l'éthique : celle de Karl Popper, avec le principe de protection de la liberté par l'État.



- Il est plus simple de choisir une épistémologie et une éthique conçues pour se compléter plutôt qu'indépendantes, car il faudra les arrimer, ce qui peut s'avérer complexe.

2.3. Description d'une première solution

La capacité de soutenir le raisonnement, voire de l'automatiser, est au coeur des capacités recherchées et des exigences à satisfaire.

- Raisonnement
 - Le raisonnement est une suite de propositions conformes à des règles de déduction convenues et fondée sur un ensemble restreint d'axiomes également convenus.
- Logique
 - La logique décrit les lois du raisonnement gouvernant les prédicats (donc les propositions).
 - Il existe plus d'une logique (logique modale, logique de description, etc.)
 - Le présent document se limite à la logique classique, bivaluée (vrai, faux), dite logique de Boole, la plus prégnante et notamment utilisée par la géométrie classique, la théorie des ensembles et la théorie relationnelle (mais non par SQL!).
- Relation
 - La relation est une représentation privilégiée du prédicat qui en facilite le calcul grâce à une algèbre appropriée [adéquate] qui sera présentée ci-après.
- Typage
 - Le typage, fondé sur la théorie des ensembles, réduit les risques d'ambiguïté, de paradoxe et d'indécidabilité dans la formulation et le calcul des prédicats.

Ce cheminement et la théorie relationnelle qui en découle sont le fruit de travaux que Edgard F. Codd entreprit au cours des années 1960 et mena jusqu'au début des années 1990. La publication séminale date de 1969.

2.3.1. Typage

- Valeur, représentation, type
 - parce qu'il faut des atomes, les identifier et les catégoriser.
- Type de base et sous-type
 - parce qu'il faut limiter l'univers du discours au (plus) juste.
- Type scalaire et non scalaire
 - parce qu'il faut exposer la structure ou pas.
- Type prédéfini et constructeur de type
 - parce qu'il ne faut pas continuellement réinventer la roue.

Pourquoi faut-il des atomes ?

Pour établir un point de départ, un niveau d'abstraction.

2.3.2. Modèle relationnel

- Concepts
 - valeur
 - type
 - fonction
 - variable \leftrightarrow état (pas d'état sans variable, pas de variable sans état)
 - procédure
 - automatisme

- Structure
 - attribut
 - tuple
 - relation
 - base (un tuple de relations)

En regard d'un univers d'intérêt (dit univers de discours):

- Un système d'information est un modèle permettant d'interpréter (de donner un sens à, d'associer une information à) des observations faites sur cet univers.
- Un prédicat est un ensemble d'observations de même « nature » (c'est-à-dire décrite par le même énoncé formulé en termes des mêmes dimensions).
- Une proposition correspond à une observation, un « fait », pouvant comporter plusieurs dimensions.
- Un paramètre correspond à une dimension de l'observation, à un « aspect » d'un fait.

En tant qu'objet de la théorie relationnelle :

- Un attribut est une paire formée d'un identifiant et un type.
- Un tuple est une paire formée d'un entête (un ensemble d'attributs) et d'une valeur (une association entre chacun des identifiants d'attribut et une valeur du type correspondant).
- une relation est une paire formée d'un entête et d'un ensemble de tuples ayant le même entête.
- une base est un tuple dont tous les attributs sont de type relation.

Transposé dans le domaine de la logique :

- Une base représente un système d'information composé de plusieurs prédicats.
- Une relation représente un prédicat (une fonction logique paramétrée).
- Un tuple représente une proposition (une fonction logique, un prédicat, dont chaque paramètre est associé à un argument).
- Un attribut est un paramètre.

Au moyen des types associés à chaque objet (attribut, tuple, relation et base), une contrainte (expression déterminant les seules valeurs admissibles de l'objet) leur est associée.

Dans la pratique, il est souvent (très) difficile d'établir un ensemble de contraintes nécessaires et suffisantes (encore et toujours Gödel). Mais on tente de s'en approcher le plus possible.

L'état du système d'information est représenté par une « variable de base » (une seule), même s'il est généralement convenu de désigner chacun de ses attributs comme une « variable de relation ».

Finalement, une variable de relation peut être définie

- soit par énumération (l'ensemble de toutes les propositions vraies et elles seules);
- soit par compréhension (l'ensemble de toutes les propositions résultant de l'évaluation d'une expression en termes d'autres attributs de la base).

2.3.2.1. Attribut

- Un attribut est un couple formé d'un identifiant a et d'un type D , noté $a:D$.
- Par abus de langage, lorsque le contexte le permet, il est usuel de désigner l'attribut par son seul identifiant; ainsi écrit-on l'attribut a .

2.3.2.2. Tuple

- Soit a_i des identifiants distincts et D_j des types, un tuple t est défini comme suit:
 - $t \triangleq (\{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}; \{(a_1,v_1), (a_2,v_2), \dots, (a_n,v_n)\})$
 - avec $\forall i: 1 \leq i \leq \deg(t) \Rightarrow \text{val}(t, a_i) \in \text{def}(t, a_i)$
- où
 - $\text{def}(t) = \{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}$ entête de t
 - $\text{def}(t, a_i) = D_i$ type de l'attribut a_i de t
 - $\text{val}(t) = \{(a_1,v_1), (a_2,v_2), \dots, (a_n,v_n)\}$ valeur de t
 - $\text{val}(t, a_i) = v_i$ valeur de de l'attribut a_i de t
 - $\deg(t) = n$ degré de t
 - $\text{id}(t) = \{a_1, a_2, \dots, a_n\}$ les identifiants d'attributs de t

Le cas $\deg(t) = 0$

Il n'existe qu'un seul tuple possible répondant à ce critère (pourquoi?):

- $t_0 = (\{\}, \{\})$

Notation simplifiée du tuple

Notation fondée sur l'ordre d'énumération des attributs, elle est souvent utilisée lorsque les identifiants d'attributs et leurs types peuvent être déterminés par ailleurs:

- $t = \langle v_1, v_2, \dots, v_n \rangle$

Notation simplifiée de l'attribut d'un tuple

La notation $t.a_i$ désigne l'attribut $a_i:D_i$ dans $\text{def}(t)$.

2.3.2.3. Relation

○ Soit a_i des identifiants distincts, D_j des types et t_k des tuples, une relation R est définie comme suit:

- $R \triangleq (\{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}; \{t_1, t_2, \dots, t_m\})$
- avec $\forall i: 1 \leq i \leq \text{card}(R) \Rightarrow \text{def}(R) = \text{def}(t_i)$

○ Où

- $\text{def}(R) = \{a_1:D_1, a_2:D_2, \dots, a_n:D_n\}$ entête de R
- $\text{def}(R, a_i) = D_i$ type de a_i de R
- $\text{val}(R) = \{t_1, t_2, \dots, t_m\}$ valeur de R
- $\text{deg}(R) = n$ degré de R
- $\text{card}(R) = m$ cardinalité de R
- $\text{id}(R) = \{a_1, a_2, \dots, a_n\}$ identifiants d'attributs de R

Le cas $\text{deg}(R) = 0$.

Il n'existe que deux relations possibles répondant à ce critère (pourquoi?):

- $R_0 = (\{\}, \{\})$
- $R_1 = (\{\}, \{t_0\})$

et elles sont très importantes en regard des opérations relationnelles, comme le zéro et le un pour les entiers!

Notation simplifiée de l'attribut d'une relation

La notation $R.a_i$ désigne l'attribut $a_i:D_i$ dans $\text{def}(R)$.

2.3.2.4. Base

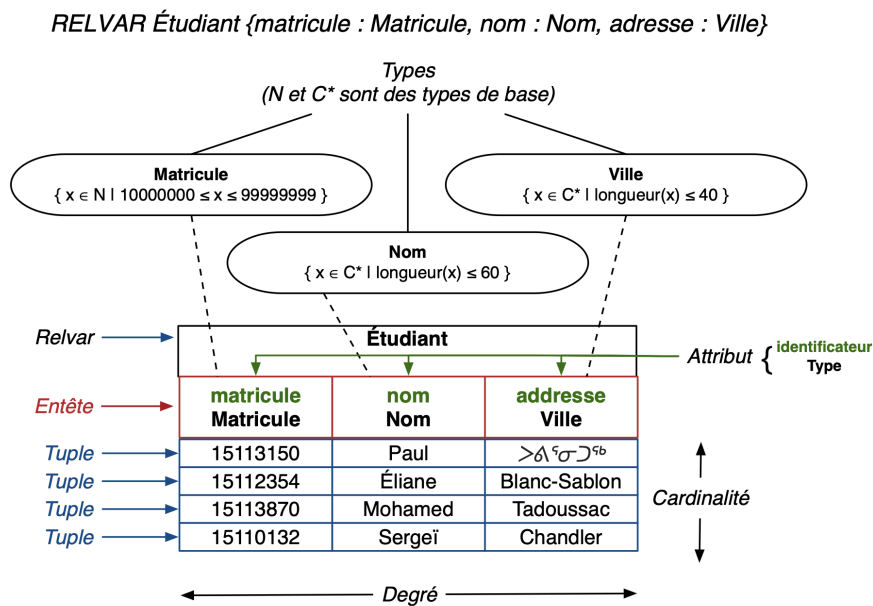
○ Soit v_i des identifiants distincts, D_j des types de relation et r_k des (valeurs de) relations, une base (de données) B est définie comme suit:

- $B \triangleq (\{v_1:D_1, v_2:D_2, \dots, v_n:D_n\}; \{r_1, r_2, \dots, r_m\})$
- avec $\forall i: 1 \leq i \leq \text{card}(B) \Rightarrow \text{def}(B, v_i) = \text{def}(r_i)$

○ Où

- $\text{def}(B) = \{v_1:D_1, v_2:D_2, \dots, v_n:D_n\}$ entête de B
- $\text{def}(B, v_i) = D_i$ type de v_i de B
- $\text{val}(B) = \{r_1, r_2, \dots, r_m\}$ valeur de B
- $\text{deg}(B) = n$ degré de B
- $\text{id}(B) = \{v_1, v_2, \dots, v_n\}$ ensemble des identifiants de variables de relation de B

2.3.2.5. Les composantes de la relation



2.3.2.6. La densification de la représentation grâce au typage

Une relation comprenant quatre tuples

<i>matricule</i> : Matricule	<i>nom</i> : Nom	<i>adresse</i> : Ville	
<i>matricule</i> : Matricule	<i>nom</i> : Nom	<i>adresse</i> : Ville	t1
<i>matricule</i> : 15113150	<i>nom</i> : Paul	<i>adresse</i> : >Δ ⁵ σ ^{5b}	
<i>matricule</i> : Matricule	<i>nom</i> : Nom	<i>adresse</i> : Ville	t2
<i>matricule</i> : 15112354	<i>nom</i> : Éliane	<i>adresse</i> : Blanc-Sablon	
<i>matricule</i> : Matricule	<i>nom</i> : Nom	<i>adresse</i> : Ville	t3
<i>matricule</i> : 15113870	<i>nom</i> : Mohamed	<i>adresse</i> : Tadoussac	
<i>matricule</i> : Matricule	<i>nom</i> : Nom	<i>adresse</i> : Ville	t4
<i>matricule</i> : 15110132	<i>nom</i> : Sergeï	<i>adresse</i> : Chandler	

La représentation compacte usuelle de cette même relation

<i>matricule</i> : Matricule	<i>nom</i> : Nom	<i>adresse</i> : Ville
15113150	Paul	>Δ ⁵ σ ^{5b}
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

2.3.3. Algèbre relationnelle

- **renommage**, $R \rho a:b$, la relation comprenant tous les tuples formés à partir d'un tuple de R dont l'attribut de nom a est remplacé par un attribut de nom b de même valeur, et rien d'autre ;
- **restriction**, $R \sigma c$, la relation comprenant tous les tuples de R satisfaisant la condition c , et rien d'autre ;
- **projection**, $R \pi x$, la relation comprenant tous les tuples formés à partir d'un tuple de R dont seuls les attributs dont le nom est parmi x ont été conservés, et rien d'autre ;
- **jointure**, $R \bowtie S$, la relation comprenant tous les tuples formés des attributs d'un tuple de R et de ceux d'un tuple de S dont les attributs de même nom sont de même valeur, et rien d'autre ;
- **union**, $R \cup S$, la relation comprenant tous les tuples de R et tous les tuples de S , et rien d'autre ;
- **intersection**, $R \cap S$, la relation comprenant tous les tuples qui sont à la fois dans R et dans S , et rien

d'autre ;

- **différence**, $R - S$, la relation comprenant tous les tuples de R qui ne sont pas dans S , et rien d'autre.
- **produit**, $R \times S$, la relation comprenant tous les tuples formés des attributs d'un tuple de R et de ceux d'un tuple de S , et rien d'autre ;

Note sur le renommage

Le statut de l'opérateur de renommage est encore discuté au sein de la communauté scientifique.

1. Il est possible de se passer de l'opérateur de renommage si on intègre une structure de catalogue à la théorie relationnelle, puisque le renommage peut alors être exprimé à l'aide des autres opérateurs relationnels appliqués aux variables de relation appropriées du catalogue.
2. L'opérateur de renommage découle d'un principe plus général, le principe de substitution. En ce sens, il n'appartient pas à la théorie relationnelle en propre, mais à la formalisation de toute algèbre.

Toutefois, le prix à payer afin d'omettre l'opérateur de renommage est une complexification non négligeable de la formalisation de la théorie et, corolairement, des modèles qui en découlent et donc des expressions relationnelles en regard de ceux-ci. Pour cela, il est usuel de le maintenir dans les opérateurs de base.

Ceci n'empêche pas pour autant l'abandon du principe plus général. Une syntaxe spécifique, semblable à celle fréquemment utilisée en mathématique pour le même usage (*soit* $x = \exp : f(x)$), peut être définie. Cet opérateur est fréquemment présent dans les langages relationnels (par exemple, en SQL **WITH** x **as** (exp) $f(x)$).

2.3.3.1. Intersection, union, différence

Intersection
 $R \cap S$

A	B
R	
S	

Union
 $R \cup S$

A	B
R	
S	

Différence
 $R - S$

A	B
R	
S	

- Ces trois opérations ensemblistes ne sont bien définies que si les entêtes des opérandes sont identiques.

2.3.3.2. Restriction, projection, jointure

Restriction
 $R \sigma \text{ cond}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Projection
 $R \pi \{A, C\}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Jointure naturelle
 $R \bowtie S$

A	B	B	C
a1	b1	b1	c1
a2	b1	b2	c2
a3	b3	b3	c3
a4	b4	b3	c4

- L'opération de projection n'est bien définie que si tous les identifiants d'attributs sont définis dans l'entête de la relation.
- L'opération de restriction n'est bien définie que si tous les identifiants d'attributs de la condition sont définis dans l'entête de la relation.

2.3.3.3. Renommage

Renommage

$R \rho A:C$

A	B		C	B
a1	b1	$\rho A:C =$	a1	b1
a2	b2		a2	b2
a3	b3		a3	b3

- L'opération de renommage n'est bien définie que si l'identifiant à changer est défini dans l'entête de la relation et que le nouvel identifiant ne l'est pas.

2.3.3.4. Panoplie minimale d'opérations

Nous retirons l'intersection des opérations de base, mais y ajoutons le renommage.

Le nombre d'opérations de base est donc limité à six :

- Trois opérations relationnelles : projection, restriction et jointure.
- Deux opérations ensemblistes : union et différence.
- Une opération structurale : renommage.

Restriction $R \sigma \text{cond}$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Projection $R \pi \langle A, C \rangle$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

Jointure (naturelle) $R \bowtie S$

A	B		B	C		A	B	C
a1	b1	\bowtie	b1	c1	$=$	a1	b1	c1
a2	b1		b2	c2		a2	b1	c1
a3	b3		b3	c3		a3	b3	c3
a4	b4		b3	c4		a3	b3	c4

Différence $R - S$

A	B
R	
S	

Union $R \cup S$

A	B
R	
S	

Renommage $R \rho A:C$

A	B		C	B
a1	b1	$\rho A:C =$	a1	b1
a2	b2		a2	b2
a3	b3		a3	b3

Réflexions

- Le produit cartésien doit-il être ajouté ?
- Le renommage, est-il vraiment une opération ?
- Quel est le noyau minimal (opérations nécessaires et suffisantes) ?
- Ce noyau minimal est-il unique ?
- Que penser du noyau $\{\langle \text{NAND} \rangle, \langle \text{REMOVE} \rangle\}$? Voir [Date2008a].

2.4. Vérification de la solution



Atelier, étape 8

Le méta-modèle répond-il aux exigences ?

Le méta-modèle est-il adéquat ?

Mêmes questions pour le modèle !

2.4.1. Satisfaction des exigences du méta-modèle

La démonstration de la satisfaction des besoins présentés à l'**étape 3** et des exigences spécifiées aux **étapes 4, 5 et 6** reprend les travaux de nombreux scientifiques (et parmi les plus grands). Travaux qui furent revus et commentés par les pairs, publiés et rendus accessibles.

Voir les remerciements et les références à la fin du présent document.

2.4.2. Adéquation du méta-modèle

- Critères absolus
 1. Validité : oui.
 2. Efficacité : oui.
 3. Cohérence : oui.
- Critères relatifs
 1. Complétude : suffisante (complétude au sens de Turing démontrée).
 2. Efficience : parmi les solutions générales connues, la plus efficiente.
 3. Évolutivité : pas moins bonne que les autres !
- Méta-critères
 1. Réfutabilité (en fonction de l'épistémologie de référence).
 2. Acceptabilité (en fonction de l'éthique de référence).

2.5. Première expérimentation

On compte sur vous pour réaliser cette expérimentation lors des ateliers !



Atelier, étape 9

- Analyse du problème posé par le cas d'études
- Définition des besoins
- Modélisation
- Spécification des exigences
- Prototype de solution
- Vérification et Validation

Vive l'APP !

2.6. Conclusions préliminaires

- En fait, il s'agit plutôt des conclusions anticipées.
- Autrement dit, les hypothèses à vérifier lors de l'expérimentation.
- Hypothèses découlant de l'analyse du méta-modèle et de celle de la littérature scientifique

contemporaine.



Atelier, étape 10

- Bilan de l'expérimentation
 - Acquis, défis, pistes
- Bilan du méta-modèle
 - Acquis, défis, pistes
- Bilan de la formation
 - Acquis, défis, pistes

2.6.1. Acquis et défis

Pourquoi présenter acquis et défis conjointement ?

Parce que de nombreux acquis théoriques tardent encore à être mis à disposition, sinon utilisés, en pratique !

Ambigüité

Une certaine amélioration grâce à l'algèbre relationnelle et aux raisonneurs, mais le problème demeure en théorie (lire notre ami Gödel) et en pratique (complexité algorithmique des algorithmes de raisonnement).

Complétude

La finitude résout théoriquement le problème de complétude... mais persiste toujours en pratique (voir complexité algorithmique et volume de données).

Données manquantes

Plusieurs solutions ont été proposées, dont deux seulement sont solides (Codd:logique4V et Date:décomposition); malheureusement, aucun langage utilisé couramment ne permet de les appliquer commodément.

Agents

Résolu en théorie — en pratique : des enjeux de complexité et d'expressivité sont toujours présents.

Axes spatiaux

Résolu en théorie — en pratique : outils disponibles, mais la complexité demeure et l'expressivité ainsi que l'efficacité demande encore des avancées significatives.

Axes temporels

Plusieurs approches intéressantes, aucune exhaustive, peu d'outils en pratique... mais qu'est-ce que le temps ?

Cohérence

Que se passe-t-il si on ajoute l'exigence suivante « Capacité de transposer les règles de pratique (tant médicales que cliniques) » dans le modèle ?

2.6.2. Pistes

- S'appuyer sur les connaissances dument documentées, établies et, de préférence, formellement démontrées.
- Réduire les sources d'ambigüité en faisant appel au principe d'universalité, aux normes et aux standards.
- Utiliser les ontologies appliquées pour décrire les prédicats et les entités auxquelles ils font référence.

3. Trois réflexions

- Du problème à la solution,
 - *en passant par les besoins, les modèles, les exigences et l'adéquation.*
- Des modèles aux données,
 - *en passant par la connaissance.*
- Du raisonnement aux relations,
 - *en passant par la logique.*

Remerciements

- Pythagore (vers -580 à -495), Socrate (vers -470 à -399),
- Platon (vers -428 à -347), Aristote (vers -384 à -322),
- Euclide (vers -330 à -270), Ératosthène (vers -276 à -194),
- Pappus (vers 290 à 350), Hypatie (vers 355 à 415),
- *Descartes (1596 à 1650), Pascal (1623 à 1662),*
- Russell (1872 à 1970), Wittgenstein (1889 à 1951), Gödel (1906 à 1978),
- Chomsky (1928...), Hoare (1934...), Floyd (1936-2001),
- Codd (1923 à 2003), Date (1941...).

Références

[Bray2002a]

Ian K. BRAY;
An introduction to requirements engineering;
Addison-Wesley (Pearson Education), Harlow (UK), 2002;
ISBN 978-0-201-76792-6.

[Codd1970a]

Edgard F. CODD;
A Relational Model of Data for Large Shared Data Banks;
Communications of the ACM, 13(6), pp. 377–387, 1970;
doi:10.1145/362384.362685.

[Codd1990a]

Edgard F. CODD;
The Relational Model for Database Management: Version 2;
Addison-Wesley Longman Publishing, Boston (MA, USA), 1990;
ISBN 0-201-14192-2.

[Date1998a]

Chris J. DATE, Hugh DARWEN;
First Edition : *Foundation for Object/Relational Database Systems: The Third Manifesto*;
Addison-Wesley Redwood City (CA, US), 1998; ISBN 0-201-30978-5.
Second Edition : *Foundation for Future Database Systems: The Third Manifesto*;
Addison-Wesley, Redwood City (CA, US), 2000; ISBN 0-201-70928-7.
Third edition : *Databases, types, and the relational model: The third manifesto*;
Addison-Wesley (Pearson Education), 2007; ISBN 0-321-39942-0.
Third revised edition : *Databases, types, and the relational model: The third manifesto*;
2014, <https://www.dcs.warwick.ac.uk/~hugh/TTM/DTATRM.pdf> (consulté le 2024-05-30).

[Date2012a]

Chris J. DATE;
Database Design & Relational Theory;
O'Reilly Media, 2012;
ISBN 978-1-449-33801-6.

[Date2014a]

Chris J. DATE, Hugh DARWEN, Nikos A. LORENTZOS;
Time and Relational Theory: Temporal Databases in the Relational Model and SQL;
Morgan Kaufmann, Waltham (MA, US), 2014;
ISBN 978-0-12-800631-3.

[Date2020a]

Chris J. DATE;
Logic and Relational Theory;
Technics Publications, Basking Ridge (NJ, US), 2020;
ISBN 978-1634628754.

[DoD2010a]

Deputy Chief Information Officer;
DoDAF — DoD Architecture Framework;
Version 2.02, U. S. Department of Defense, 2010;
<http://dodcio.defense.gov/Library/DoD-Architecture-Framework/> (consulté le 2024-05-30).

[Hull2011a]

Elizabeth HULL, Ken JACKSON, Jeremy DICK;
Requirements Engineering;
Third Edition, Springer-Verlag, London (UK), 2011;
ISBN 978-1-84996-405-0.

[Jackson2001a]

Michael JACKSON;
Problem frames: analysing and structuring software development problems;
Addison-Wesley, 2001;
ISBN 978-0-201-59627-4.

[Khnaisser2015a]

Christina KHNAISSER, Luc LAVOIE, Hassan DIAB, Jean-François ETHIER;
Architecture et méthode de construction d'entrepôts de données au coeur des systèmes informationnels régionaux de santé;
Rapport de recherche, Groupe de recherche interdisciplinaire en informatique de la santé (<http://GRIIS.ca>),
Université de Sherbrooke, Sherbrooke (QC, CA), juin 2015.

[Khnaisser2015b]

Christina KHNAISSER, Luc LAVOIE, Hassan DIAB, Jean-François ETHIER;
Data Warehouse Design Methods Review: Trends, Challenges and Future Directions for the Healthcare Domain;
ADBIS 2015 - 19th East-European Conference on Advances in Databases and Information Systems, Poitiers (FR);
DOI 10.1007/978-3-319-23201-0_10.

[Popper1935a]

Karl POPPER;
Logik der Forschung;
Zur Erkenntnistheorie der modernen Naturwissenschaft, Julius Springer, Wien 1935.
Traduction française : *Logique de la recherche*; Payot, Paris, 1973.

[Popper1963a]

Karl POPPER;
Conjectures and Refutations: The Growth of Scientific Knowledge;
Routledge & Kegan Paul, London, 1963; ISBN 0415043182.
Traduction française : *Conjectures et réfutations*; Payot, Paris, 1985.

[Popper1972a]

Karl POPPER;
Objective Knowledge: An Evolutionary Approach;
Clarendon, Oxford, 1972, Revised edition 1979; ISBN 01987502421972.
Traduction française : *La connaissance objective*; Aubier, Paris, 1991.

Produit le 2025-05-26 09:36:11 UTC



École interdisciplinaire en numérique de la santé