

Bases de données : d'hier à demain

Autor(en): **Spaccapietra, S.**

Objektyp: **Article**

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

Band (Jahr): **91 (1993)**

Heft 3: **Computer Aided Engineering**

PDF erstellt am: **11.09.2024**

Persistenter Link: <https://doi.org/10.5169/seals-234940>

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Bases de données: d'hier à demain

S. Spaccapietra

Peut-on espérer avoir des SGBD «orientés utilisateurs»? Le laboratoire des bases de données de l'EPFL pose cette idée comme principe de base de sa recherche. Son objectif est de développer une interface utilisateurs qui permette à ceux-ci d'exprimer leurs interactions avec la base de données, sans avoir à se soucier de la technologie mise en jeu.

Darf man auf «benutzerorientierte» Datenbanken hoffen? Das Laboratorium «Datenbanken» der ETH Lausanne erhebt diese Idee zum Grundprinzip seiner Forschung. Das Ziel ist, eine Benützeroberfläche zu entwickeln, die dem Benützer einen Dialog mit der Datenbank erlaubt, ohne sich um die dahinterstehende Technologie kümmern zu müssen.

Les systèmes de gestion de bases de données (SGBD) assurent une gestion intégrée de toutes les données nécessaires au fonctionnement d'un ensemble d'applications. L'intérêt premier de ces systèmes est de garantir le maintien de la cohérence des informations tout en permettant une utilisation partagée. C'est donc une approche qui se veut à la fois très souple et très sûre. Très souple, car tout utilisateur peut bâtir ses applications comme il l'entend, dans la limite des autorisations d'accès dont il dispose. Très sûre, car il n'est pas possible d'enfreindre les règles de cohérence établies d'un commun accord et implantées dans le SGBD.

Ces avantages expliquent la diffusion de la technologie des bases de données. Née dans les années 60, celle-ci a connu une véritable explosion dans les années 80. Elle débordait aujourd'hui largement son domaine traditionnel (gestion d'entreprise) et investit avec succès les applications les plus variées: géomatique (cartographie, gestion du territoire, ...), productique, bureautique, CAO, ..., y compris l'informatique elle-même (génie logiciel).

Cet essor s'explique facilement. D'une part par une demande pressante due aux nouvelles contraintes de compétitivité qui pèsent sur les entreprises dans tous les secteurs économiques. D'autres part par une plus grande disponibilité de l'outil (à prendre dans le sens: plus facilement utilisable). La première génération des SGBD résultait essentiellement d'un saut qualitatif dans les techniques de gestion de fichiers. Leur mode de fonctionnement relevait des techniques informatiques (gestion de pointeurs, gestion du stockage des données, ...). Les outils produits (IDS,

IDMS, DMS/1100, TOTAL et beaucoup d'autres), sont destinés à des informaticiens, et par là même à diffusion restreinte.

Cacher les techniques d'implémentation

A contrario, la deuxième génération (les SGBD relationnels: DB2, ORACLE, INGRES, SYBASE, INFORMIX, etc.) est partie de l'idée de cacher les techniques d'implémentation derrière une interface utilisateur où ne seraient visibles que les structures logiques des données. Les interactions utilisateurs/SGBD s'en trouvaient ipso facto sensiblement simplifiées. Le dialogue était donc abordable par des non-spécialistes. Le principe posé, restait à choisir la structure de données offerte aux utilisateurs-gestionnaires. Le critère qui l'emporta à l'époque est celui de la simplicité: une structure plate, la relation (ou table), et unique. Facile à comprendre, facile à gérer, facile à formaliser. Cela permit de construire des SGBD complets et performants, mais également et à peu de frais des mini-SGBD à fonctionnalités réduites (DBASE, OMNIS, 4ème Dimension, ...) pour le marché des micro-ordinateurs. C'est assurément ce dernier marché qui est à créditer pour le «boom» des systèmes relationnels.

Il fallait, pourtant, que la simplicité se paie quelque part, car la structure des données n'est pas, dans la réalité, aussi élémentaire qu'une relation. Dès que l'on cherche à donner une représentation fidèle d'un objet du monde réel (un employé, par exemple) on est amené à lui associer plusieurs propriétés (numéro, nom, prénoms, formation, qualifications, ...), dont certaines seront décrites à leur tour par un ensemble de propriétés composantes (la formation sera décrite par un ensemble de diplômes, dont on voudra connaître le titre, la date d'obtention – année et mois –, l'école, ...). Cela se traduit naturellement

par une structure arborescente (i.e.: des décompositions successives), bien loin de la structure plate d'une relation. De plus, la réalité d'une application comporte, au-delà des objets, nombre de liens entre ces objets. Ainsi, pour cet employé, on voudra savoir dans quel service il travaille, à quels projets il est affecté, de quelle machine il est responsable, etc. ... Les systèmes relationnels traitent les liens comme des propriétés supplémentaires des objets (on ajoutera aux propriétés de l'employé l'identifiant du service dans lequel il travaille), ou comme des objets-relations supplémentaires (on créera une relation «affectation» entre employés et projets). Ce faisant ils dénaturent l'information représentée et, en mettant tout dans le même sac, ils perdent le système intuitif de repérage qui permet à l'utilisateur d'appréhender une structure d'informations complexe.

L'utilisation d'un SGBD relationnel implique donc un effort de modélisation important pour arriver à transposer les données des applications en des structures admises par le système. Cette tâche, la conception de la base, prend beaucoup de temps et demande des compétences spécifiques. D'où l'essor des méthodes de conception (MERISE, YOURDON, SADT, IDA, ...): démarches méthodologiques et outils développés pour guider et aider le concepteur.

Permettre un meilleur contrôle des opérations

Ce hiatus entre réalité et logiciels ne pouvait durer indéfiniment. Les systèmes orientés objets (OO) qui se développent aujourd'hui offrent une première réponse. Cette troisième génération (O2, ONTOS, VERSANT, ...) restaure le concept d'objet en tant que tout unitaire, quelle que soit la complexité des propriétés qui le décrivent. De plus, les SGBD-OO innovent en permettant d'associer à chaque type d'objet, en plus de la description de ses propriétés, la description des opérations permises sur les objets de ce type. De cette façon, une connaissance plus complète des objets est transmise au logiciel, qui en assure alors la gestion. Cela décharge les applications et permet un meilleur contrôle des opérations. Les SGBD-OO représentent ainsi une avancée substantielle dans les services rendus aux utilisateurs et il ne fait pas de doute qu'ils connaîtront, à terme, un succès similaire à celui des SGBD relationnels.

L'orientation objets, toutefois, traduit ses origines: les langages de programmation. En d'autres termes, elle est plus une façon d'implémenter qu'une façon de raisonner au niveau d'abstraction qui est celui des applications. Ainsi, par exemple, sa représentation des liens tient plus du pointeur

(de A vers B) que de l'association (entre A et B). La description des opérations fait appel à un langage de programmation, plutôt que de spécification. L'encapsulation, qui oblige à associer une opération à un seul type d'objet, fait primer l'efficacité du développement sur l'élégance de la modélisation. En résumé, les SGBD-OO enrichissent les SGBD, et les rapprochent quelque peu des utilisateurs, mais s'en tiennent encore à distance. En fait, ce rapprochement est un effet de bord de l'objectif fondamental, qui est de faciliter le développement et la mise au point du logiciel.

Prendre en charge les utilisateurs

Peut-on espérer combler ce fossé et avoir enfin des SGBD «orientés utilisateurs»? Certainement. Mais compte tenu des impératifs industriels et du marché, il est plus réaliste de penser que des outils CASE viendront jouer les intermédiaires entre gestionnaires et logiciels. Cette approche est déjà en route, bien que de façon embryonnaire. Le laboratoire de bases de données principe de base de sa recherche.

che. Notre objectif est précisément de développer une interface utilisateurs qui permette à ceux-ci d'exprimer toutes leurs interactions avec la base de données, sans avoir à se soucier de la technologie mise en jeu. Dans cette interface n'apparaît que l'expression des besoins des applications, l'outil se chargeant de répercuter les actions vers le SGBD sous-jacent (relationnel ou orienté-objets). Concrètement, la prise en charge des utilisateurs est assurée par un ensemble d'outils coopérants, chaque outil étant spécifique d'une tâche: définition du schéma de la base, chargement des données, interrogation, etc. Le dialogue se fait par la voie graphique, par manipulation directe d'objets à l'écran, ou textuelle, au choix des usagers.

La spécification des outils s'appuie sur les résultats de recherche. Ainsi, le modèle utilisé pour la description des données cumule les avantages de l'approche objets avec ceux de l'approche entité-relation, bien connue des utilisateurs d'outils et méthodes de conception. En élaborant le modèle, le laboratoire a également conçu les opérateurs de manipulation qui lui cor-

respondent. C'est ce qui nous met en mesure de spécifier une interface complète et cohérente, qui permet à l'utilisateur de raisonner avec les mêmes concepts lors de la conception et lors de l'exploitation de la base (ce qui n'est pas le cas des outils CASE existant). Nous développons également les outils méthodologiques pour la phase de conception, avec une approche novatrice qui repose sur une conception décentralisée. De cette façon, chaque service d'une entreprise peut concevoir en toute liberté son schéma de la partie de la base de données qu'il va utiliser, l'outil se chargeant ensuite d'intégrer les différents schémas partiels en un schéma global d'entreprise. Le gain en autonomie et en flexibilité est évident. Un prototype existe pour les outils de définition de schéma et interrogation de la base. Repris par une société privée, il pourrait donner un outil très compétitif sur le marché CASE pour les bases de données.

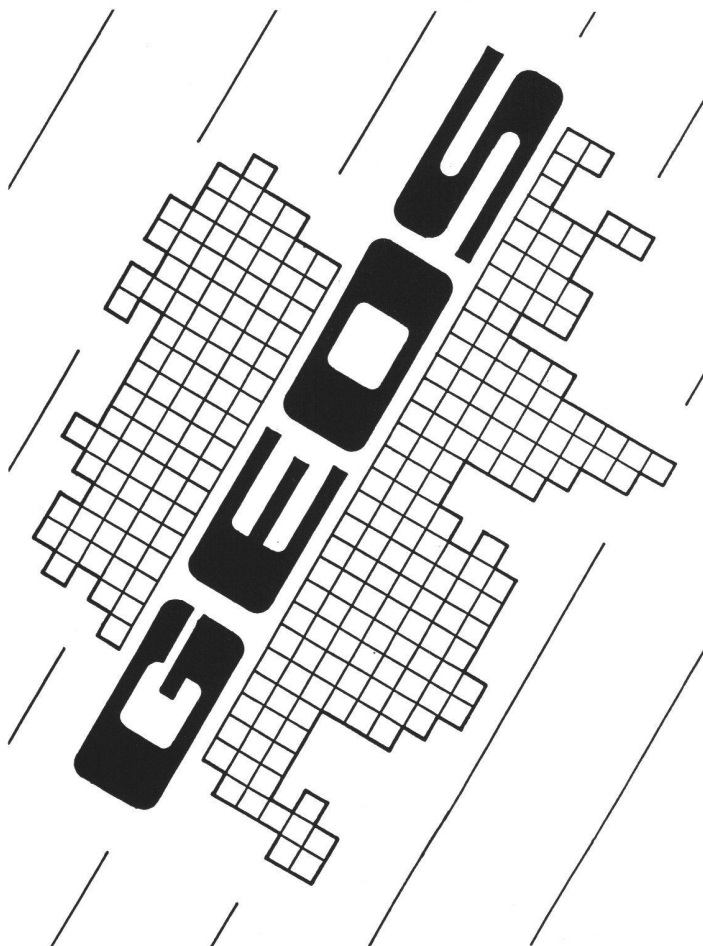
Adresse de l'auteur:
Prof. Dr. Stefano Spaccapietra
Laboratoire de bases de données
EPFL DI-LBD
CH-1015 Lausanne

a/m/t software service ag

Obergasse 2a

8400 Winterthur

Tel. 052 213 23 13



GEOS 4

Die Lösung für die RAV

- Netz- und Punktberechnung mit Qualität und Zuverlässigkeit
- Mutationsverwaltung
- interaktive Graphik
- thematische Ebenen

La solution pour la REMO

- réseaux polygonométriques et points de détail avec qualité et fiabilité
- gestion des mutations
- graphique interactive
- niveaux thématiques