Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

Herausgeber: Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

Band: 80 (1989)

Heft: 3

Artikel: Knowledge Craft : présentations et applications

Autor: Braun, F.

DOI: https://doi.org/10.5169/seals-903635

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. 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. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

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. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

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. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 09.12.2025

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

Knowledge Craft: présentations et applications

F. Braun

In diesem Beitrag wird die KI-Entwicklungsumgebung Knowledge Craft vorgestellt. Ihre Hauptelemente und Zusatzfunktionen samt deren Möglichkeiten und Interaktionen werden beschrieben. Die Stärken und Schwächen von Knowledge Craft sowie eventuelle Verbesserungsmöglichkeiten kommen ebenso zur Sprache wie Erfahrungen auf verschiedenen Anwendungsgebieten.

Knowledge Craft, un environnement de développement d'Intelligence artificielle, est ici présenté, en explicitant ses principaux éléments, les fonctionalités annexes dont on soulignera les apports et leurs interactions. Sa position dans l'offre actuelle d'outils d'intelligence artificielle est discutée, en mettant en relief ses points forts et les améliorations possibles. Pour chaque composant de l'outil, des exemples d'applications l'ayant particulièrement exploité sont cités. Enfin, quelques-uns des domaines typiques d'utilisation de Knowledge Craft sont référencés.

Adresse des Autors
F. Braun, Ing. civil de l'E.N.S.T., Ma.Sc.,
Synlogic AG,
Hauptstrasse 34, 4102 Binningen.

Lisp est connu comme un langage adapté au développement d'applications dont une caractéristique est de manipuler des symboles autres que numériques. Cependant, durant les années 70, d'autres modes de représentation des connaissances ou langages ont été proposés, et notamment:

- les moteurs d'inférences
- la représentation par objet et les langages orientés objets
- la programmation logique.

Ces modes permettent de représenter efficacement certains aspects de problèmes comme:

- l'énoncé d'une relation de causalité et la représentation par règles de production dans un moteur d'inférence
- l'existence de concepts ou d'entités ayant des relations statiques (hiérarchiques, d'appartenance...) ou dynamiques (comportement réflexe, correlation de valeurs...)
- l'exploration de graphes ou d'arbres et la programmation logique.

Ressentant la complémentarité des outils existants, certains milieux universitaires ont intégré ces différents modes de programmation avec *Lisp* comme langage de base, au début des années quatre-vingt. Parmi eux, l'université de *Carnegie Mellon* a donné naissance à un environnement intégré de développement d'applications qui, sous sa forme commerciale, s'appelle *Knowledge Craft*.

Cet article propose de présenter Knowledge Craft, de montrer, sous la forme d'exemples et de références à des applications, ses principaux points d'intérêts et de discuter sa position dans l'offre en outils de développement en Intelligence artificielle.

Présentation de Knowledge Craft

Principaux composants de Knowledge Craft

Basé sur Common Lisp [1] Knowledge Craft [2] inclut les outils de programmation suivants (fig. 1):

- CRL, un langage orienté objet à base de schémas [3]
- CRL-OPS, une implémentation de OPS-83, un langage à base de règles [4])
- CRL-Prolog, une implémentation de Prolog [5].

En plus de ces trois outils intégrés, Knowledge Craft inclut des fonctionnalités complémentaires comme:

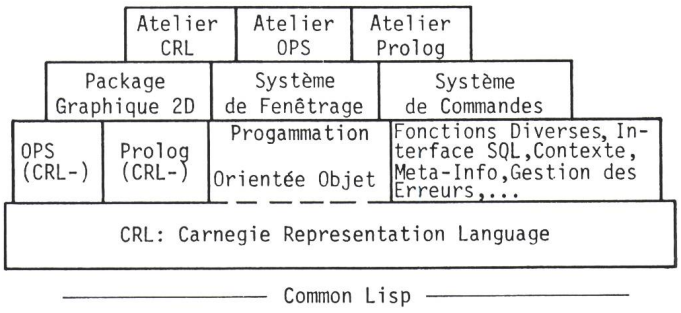

Figure 1 Organisation Générale de Knowledge Craft

- les *contextes* qui permettent des raisonnements isolés au sein de l'exécution d'une application
- la *gestion de files d'attente*, notamment pour les applications de simulation
- la *gestion de méta-connaissances*
- des *fonctions de prise en charge des erreurs*.

L'intérêt d'un tel outil, en plus des langages et fonctionnalités rassemblés mentionnés ci-dessus, consiste aussi en l'intégration d'outils d'interface adaptés aux moyens de programmation sous-jacents. On peut notamment citer:

- l'éditeur de texte ZMACS, surensemble de l'éditeur standard Lisp, adapté à l'édition de programmes CRL
- un éditeur de graphe, pour l'édition d'objets CRL et des relations entre objets
- un atelier de programmation OPS
- un atelier de programmation Prolog
- des fonctions de création d'interface conviviaux incluant un système de fenêtrage, un système de gestion de commandes de haut niveau (menus, souris, éditeur ligne...) et des possibilités de graphisme.

CRL

Le langage orienté objet CRL est un sous-ensemble de programmation de Common Lisp, et c'est aussi le langage d'implémentation de Knowledge Craft. Une caractéristique notable du langage orienté objet CRL consiste en sa représentation unifiée des concepts de classe, d'objet et de relation sous la forme de schémas. On s'affranchit ainsi de la trilogie classique de relations «est une spécialisation de», «est une instance de» et «fait partie de». CRL a servi de base à de nombreuses applications et projets d'étude, comme on le verra plus tard. Un exemple en génie chimique est fourni dans [6].

CRL-OPS

Extension de OPS-83, CRL-OPS permet d'inclure des schémas CRL dans les règles, combinant ainsi Lisp, CRL et OPS. Son efficacité est assurée par l'utilisation concomittante d'un algorithme d'exécution, dit du *Rete*. A cause de son efficacité, OPS a été parfois considéré comme *l'assembleur de l'Intelligence artificielle*. Son extention ici, en fait un langage de haut niveau. OPS a été utilisé dans des applications comme XCON et XSEL de Digital Equipment Corp. Autre exemple, CRL-OPS a permis de codifier des règles de gestion du trafic ferroviaire [7].

Tableau I
Comparaison des représentations par prédicats et objet

Représentation par objet	Représentation par prédicats
{{homme	
est un mammifère	est un(homme, mammifère)
bras 2	bras(homme, 2)
jambes 2	jambes(homme, 2)
position xy	position(homme, x, y)
...}}	

CRL-Prolog

L'intégration de Prolog à Lisp et CRL, permet, en plus de ses intérêts intrinsèques connus [5]

- d'unifier une variable Prolog au résultat d'une évaluation d'une fonction écrite en CRL
- d'enrichir la représentation par prédicats par la représentation objet (voir tableau I).

De plus, l'héritage multiple et filtré permet une richesse d'information syncopée impossible directement en Prolog.

A l'inverse, la concision de Prolog, grâce au principe intégré de résolution (moteur d'inférence), à ses notions particulières de variables, de retour-arrière, à l'écriture très déclarative des connaissances, étend les possibilités globales de l'outil. Un exemple de complémentarité consiste à modéliser une base de données sous forme d'objets, puis d'interroger et effectuer des déductions grâce aux inférences Prolog.

Les contextes

La notion de contexte est la version logique et généralisée de la segmentation matérielle en pages. C'est donc une spécification d'un univers de travail dans lequel les modes de programmation ont lieu. On peut créer, même dynamiquement, une hiérarchie de contexte. Toute donnée ou programme d'un contexte est visible dans ses contextes descendants, à moins qu'une redéfinition n'ait lieu dans l'un d'eux. Les conditions de règles OPS ou Prolog peuvent spécifier les contextes de validité. Les contextes sont utilisés pour l'étude simultanée de plusieurs cas, l'étude d'option, l'exploration par suppositions etc. Ils sont aussi utilisés dans les problèmes où la solution ne peut être imaginée à l'avance, comme en configuration, et les systèmes de maintenance de la cohérence [8]. On peut voir une illustration de l'utilisation de contextes dans un système d'aide au commandement [9].

La gestion des files d'attente

Cette fonctionnalité permet une approche cognitive des problèmes d'ordonnancement, de planification et de gestion de ressources. Elle permet aussi, par intégration avec les autres outils, de fournir un moyen de représenter et de manipuler des données temporelles. Knowledge Craft a été la base de plusieurs réalisations d'ordonnancement, dont une notamment pour une usine sidérurgique [10].

La gestion des méta-connaissances

Knowledge Craft permet la création de méta-schémas stipulant le comportement d'une propriété d'un objet. On spécifie ainsi des comportements réflexes, des filtrages de valeurs associées à la propriété, des valeurs par défaut, des explications associées et des commentaires. Ces moyens et les méta-niveaux de programmation ont été particulièrement utilisés dans un projet de robotique à base de systèmes experts coopérants [11].

Positionnement de Knowledge Craft parmi les outils d'Intelligence artificielle

Positionnement et points forts

Knowledge Craft se situe en fonctionnalités et en prix dans la classe supérieure des outils d'Intelligence artificielle. C'est pourquoi il a été utilisé dans de nombreux domaines et correspond à des besoins récemment constatés: utilisation d'outils hybrides et génériques, intérêt des approches par prototypage et leur adaptation aux

problèmes attaqués par l'Intelligence artificielle:

a. Besoin d'outils hybrides: Les problèmes d'Intelligence artificielle incluent des modèles de connaissances et des raisonnements complexes, impliquant souvent la disposition de tous les moyens actuellement industrialisés.

b. Besoin d'outils génériques: En plus de la flexibilité exceptionnelle de Lisp, l'adjonction d'autres modes de programmation facilite la construction d'un ensemble de concepts ou d'un environnement adapté à un domaine: On peut codifier ainsi une partie du savoir-faire d'une entreprise et assurer une compatibilité des réalisations. Cet aspect se retrouve dans le point suivant.

c. Approche par prototypage: Le développement d'applications en Intelligence artificielle nécessite en général une approche graduée par une maquette, un prototype ou parfois une réalisation partielle. La richesse fonctionelle de Knowledge Craft, par rapport aux outils meilleurs marchés, ne jouent plus toujours dans ces cas là en sa faveur, sauf si les développeurs sont expérimentés. Par contre, si un projet a des extentions, un outil puissant permet une uniformisation de l'écriture aux différents stades de conception, la réutilisation de codes et une sécurité accrue de la faisabilité.

Quelques limitations

a. Complexité: Knowledge Craft, comme les outils de cette classe, nécessite la maîtrise de Lisp et une phase d'acquisition assez longue. Aucun environnement existant n'a d'offre graduelle, à part quelques outils simples. Une possibilité de programmation à plusieurs niveaux, reflétant la complexité de la réalisation et l'expérience des développeurs, faciliterait l'acquisition graduelle pour une entreprise de son savoir-faire en Intelligence artificielle.

b. Isolation: Après une période d'offre limitée en support matériel, Knowledge Craft a été porté sur plusieurs plate-formes très utilisées dans l'industrie et prochainement sur PC. D'autre part, Knowledge Craft commence à être muni de modules d'interfaces avec des bases de données relationnelles, grâce à CRL-SQL.

Les applications

Knowledge Craft a été récemment utilisé pour des systèmes opérationnels divulgués dans des domaines comme la configuration (système de configuration de centraux téléphoniques), la planification et l'allocation de ressources (système de gestion de traffic ferroviaire), le contrôle de processus industriels (ordonnancement de contrôle d'une usine sidérurgique), cela aux Etats-Unis et en Europe. De plus, Knowledge Craft a été utilisé comme support de recherche ou d'élaboration de prototype préopérationnels dans d'autres domaines comme la finance, l'aéronautique, avec des systèmes d'aide à la conception, l'aide au commandement et les diagnostics de panne dans divers domaines.

Conclusion

Une revue des outils proposés dans l'environnement Knowledge Craft a permis d'y associer des éléments d'intérêt de l'Intelligence artificielle. On peut espérer une diffusion de ceux-ci dans toutes les branches de l'informatique. Ceci a déjà lieu comme le montre la recherche, par exemple, dans les systèmes d'exploitation adaptés aux réseaux [11]. La technologie actuelle, illustrée par Knowledge Craft, et les réalisations existantes opérationnelles sont suffisamment convaincantes pour prévoir une multiplication de celles-ci, en utilisant des outils de plus en plus éloignés des langages de base.

Literatur

[1] *G.L. Steele:* Common lisp: The language. Bedford/Massachusetts, Digital Press, 1984.

[2] Manuel de réference de Knowledge Craft version 3.2. Paris-La Défence, Société GSI-Tecsi, 1988.

[3] *D. Colnet e.a.:* Les langages orientés objets. CRIN 86-R-077. Nancy/France, Centre Régional Informatique de Nancy (CRIN), 1986.

[4] *C. Forgy* and *J. McDermott:* Ops, a domain-independent production system language. Fifth International Joint Conference on Artificial Intelligence 1977, p.933...939.

[5] *W.F. Clocksin* and *C.S. Mellish:* Programming in prolog. Second edition. Berlin a.o., Springer-Verlag, 1984.

[6] *P. Bourseau* e.a.: Intelligence artificielle et génie des Procédés. Informations Chimie - (1987)282, avril, p. 221...224.

[7] *J. Bentolila* et *R. Moulin:* Gespi: Un système expert de gestion prévisionnelle du trafic dans une grande gare. Congrès AFCET (Association française pour la cybernétique économique et technique) Actes des Journées d'Avignon 1988.

[8] *J. Doyle:* A truth maintenance system. Artificial Intelligence 12(1979)-, p. 231...272.

[9] *M. Babin, M. Gately* and *M. Sullivan:* Fresh: A naval scheduling system. Sixth National Conference on Artificial Intelligence, Los Altos/California, 1987; p. 196...201.

[10] *R.R. Schreiber:* Gainning the upper hand with AI. CIM Technology -(1987)5.

[11] *B. Draa Chaib* and *P. Millot:* Assignment of roles among cooperative distributed problem solving systems. Proceedings of the 19th ASME International Computers in Engineering Conference and Exhibition, San Francisco/California, 1988.

[12] INRIA. Bulletin de la liaison de la recherche en informatique et en automatique (1988)119.