

Représentation des Connaissances dans un Langage Visuel Typé : une Première Approche *

Florence Dupin de Saint-Cyr
IRIT, Université de Toulouse,
Toulouse, France
Florence.Bannay@irit.fr

Denis Parade
SCÉNARIO INTERACTIF,
Toulouse, France
infos@scenario-interactif.com

19 juin 2018

Résumé

Cet article présente un ensemble de principes qu'un langage visuel de représentation des connaissances doit satisfaire. Nous avons introduit quatre principes formel (accessibilité, navigation, efficacité d'accès, existence d'un point d'entrée) et sept postulats en langage naturel visant à contraindre la cohérence, compacité, navigabilité du langage par rapport aux attributs visuels (position, forme et couleur).

Après une présentation du langage MOT, nous introduisons le langage VTL (Visual Typed Language) qui en est inspiré. VTL est un langage visuel typé qui satisfait la plupart des principes que nous proposons.

Nous avons séparé les éléments du langage VTL en trois catégories : *entités*, *actions* et *conditions*. Les expressions de chaque catégorie sont soit des éléments génériques soit des instances. Ils sont reliés entre eux par des attributs dépendant de leur catégorie, ces liens permettent la navigation.

Enfin, nous avons réalisé une maquette interactive à l'aide du logiciel XMind, initialement dédié aux cartes mentales.

Mots clés : langage visuel, perception visuelle humaine, interactivité, navigation par zoom avant/arrière

1 Introduction

De nos jours, la transmission des idées passe principalement par la langue écrite. Comme tout langage, l'écriture doit répondre à une syntaxe et à des règles grammaticales précises. Par convention, et quelle que soit la langue utilisée, des mots composent des phrases et sont posés séquentiellement sur des lignes qui vont également guider l'œil lors de la lecture. Même si cela nous permet d'exprimer des idées complexes et subtiles, certaines représentations vont

*Cette version est un document de travail, l'article est publié dans les actes de IC'2018.

au-delà du mode linéaire et ajoutent une distribution spatiale à l'information, éventuellement sur un support interactif, ou via une approche combinant une vue globale et une vue détaillée.

Pour aller au delà de l'utilisation du texte, il y a près d'un siècle, Otto Neurath [15] a entrepris des recherches pour définir un langage graphique universel. Ce qui l'a conduit à Isotype (Système International de Typographie Picture Education). La sémiotique des pictogrammes était théorisée afin de définir une communication visuelle par des symboles graphiques compris par tous. Cette idée d'utiliser des images pour représenter et transmettre des informations a aussi été largement exploitée dans différents domaines. En effet, les pictogrammes sont des vecteurs d'information très efficaces qui sont utilisés dans les panneaux de signalisation routière par exemple. Selon Teboul [29], les êtres humains ont la capacité de traduire immédiatement une forme graphique en un élément sémantique. De plus, la perception visuelle humaine est bien adaptée pour appréhender une situation, reconnaître un lieu et par extension une représentation graphique dans son ensemble. Notre cerveau est en effet particulièrement efficace pour traiter rapidement une information visuelle (on admet¹ souvent que 90% des informations qui arrivent au cerveau sont sous cette forme). Aujourd'hui, il a été montré que les informations présentes sur Internet et les réseaux sociaux ont beaucoup plus de chances d'être comprises, mémorisées et partagées si elles comportent des éléments visuels (animés ou non).

Il existe déjà beaucoup de représentations visuelles utilisées dans la vie de tous les jours : les cartes mentales introduites par Tony Buzan [4], les cartes topiques [21], les cartes conceptuelles [16] les diagrammes de Venn [16], les frises chronologiques, les organigrammes de programmation [11], les cartes géographiques, etc. Elles ont toutes des aspects intéressants : les diagrammes de Venn sont faciles à comprendre. Les frises chronologiques et les cartes géographiques ont un fort ancrage culturel donc sont également compris immédiatement. Les organigrammes de programmation apportent une dimension ordonnée (temporelle et/ou hiérarchique) et ne sont pas une simple représentation statique. Les cartes mentales offrent une grande liberté de création, sont faciles à mettre en œuvre, et facilitent la mémorisation. Quant aux cartes conceptuelles, elles permettent à l'utilisateur de décrire un mécanisme ou une procédure sans ambiguïté. Cependant, elles ont toutes quelques inconvénients : pour les cartes mentales et les cartes topiques, il peut y avoir des ambiguïtés dans l'utilisation des mots-clés et des relations que l'utilisateur est libre de définir sans aucune contrainte. Les cartes conceptuelles sont graphiquement assez pauvres, et le sujet principal est rarement mis en évidence. Une carte conceptuelle mal composée devient rapidement illisible. Les diagrammes de Venn sont d'une utilisation très limitée, peut-être parce qu'ils sont trop simples. Les frises chronologiques nécessitent d'avoir un support physique de la bonne longueur. Il n'y a aucune possibilité de zoomer ou de modifier l'échelle linéaire. Les organigrammes exigent la connaissance des formes de base et n'utilisent pas de couleurs ou de dessins.

1. Malheureusement, autant que nous le sachions, cette affirmation est toujours utilisée sans aucune référence à une étude scientifique sur ce sujet...

Souvent, les projections cartographiques déforment la réalité.

Il existe également beaucoup de langages de représentation visuelle moins utilisés du grand public car destinés à des applications très spécifiques, la plupart du temps informatiques. C'est le cas par exemples des « graphes conceptuels » [26] destinés au codage visuel de phrases et de discours dont on veut capturer la logique, des méthodes SADT (aussi appelée IDF0) [5, 13], ODT [24] puis UML [6, 25] pour la conception de systèmes d'information et plus généralement de tous les formalismes de réseaux sémantiques (voir [27] pour un panorama). Notons aussi l'existence de langages permettant de représenter des hiérarchies comme OWL [12], ou à la cartographie des connaissances des organisations (avec entre autre le besoin de localiser les connaissances critiques) comme GAMETH [9, 10]. Aucun de ces langages n'a pour objectif principal d'être facilement appréhendé par un lecteur humain, ils demandent donc tous un apprentissage plus ou moins long.

Afin de surmonter les inconvénients des représentations visuelles existantes, nous avons commencé une démarche de formalisation de principes permettant de caractériser une représentation visuelle intuitive et efficace. Nous pensons qu'en imposant des principes rationnels basés sur des fondements en psychologie cognitive, nous pourrions certifier ou non qu'un langage visuel est adapté à la perception et compréhension humaine. Cette première étape nous a amené à définir formellement quatre principes et à en exprimer huit autres sous forme de postulats en langage naturel. De plus, nous proposons un nouveau langage appelé VTL (Visual Typed Language), langage de représentation typé visuel, qui a été créé dans l'optique de satisfaire ces postulats. Ainsi, VTL combine l'utilisation de mots-clés avec des icônes, des images, des schémas et des liens, qui aideront à saisir rapidement le sens de ce qui est exprimé. Cette combinaison d'éléments est liée à la théorie du double codage [18] : le codage d'un stimulus de deux façons différentes augmente la chance de s'en rappeler par rapport au codage de ce stimulus d'une seule façon. Notre idée est d'enrichir le modèle des cartes mentales afin d'obtenir un schéma qui admet une traduction automatique non ambiguë, qui se base sur l'aspect visuel afin de représenter les propriétés des objets et leurs liens. Notre objectif est que l'existence de cette traduction unique puisse constituer un moyen de comprendre, de raisonner et de décider visuellement.

Une version anglaise de cet article (moins développée par rapport à l'état de l'art) est publiée dans [7].

2 Principes associés à un langage de représentation visuel

Nous allons considérer qu'un langage visuel définit un ensemble d'expressions possibles, ces expressions doivent contenir des informations élémentaires reliées selon le formalisme du langage visuel. Plus précisément, nous considérons un ensemble S de symboles visuels, un ensemble W de mots anglais (les éléments de

l'ensemble $S \cup W$ sont appelés « items », et un ensemble C de connecteurs (avec leur arité). Une expression e d'un langage visuel L est une combinaison d'items (éléments de S et de W) à l'aide d'un ensemble C de connecteurs. Les items de e sont notés $items(e)$. La particularité d'un langage visuel est que toute expression e est associée à des attributs visuels tels que forme/couleur/position qui sont attachés à chacun de ses items et également à chacun de ses connecteurs. Nous notons $c(i, j) \in e$ le fait que les deux items i et j sont reliés par le connecteur c dans l'expression e . L'existence d'un chemin² de longueur k de i à j dans e est notée $path_k(i, j) \in e$. La distance entre deux items dans e est la longueur du plus court chemin entre ces deux items : $dist_e(i, j) = \min\{k \mid path_k(i, j) \in e\}$

De plus, toute expression e d'un langage visuel a au moins un item qui est considéré visible d'emblée, cet item s'appelle une entrée de e , l'ensemble des entrées de e , est noté $entrySet(e)$.

Afin de formaliser l'efficacité d'accès aux items d'une expression, nous définissons la mesure suivante :

Définition 1 *L'inefficacité d'accès dans une expression e est le pourcentage obtenu en divisant la plus grande distance à parcourir pour atteindre un item de e depuis une entrée quelconque de e , par le nombre total d'items dans e . $\forall e \in \mathcal{L}$,*

$$\mathcal{I}(e) = \frac{\max_k \{x \in entrySet(e), y \in items(e), dist_e(x, y) = k\}}{|items(e)|}$$

La mesure d'inefficacité d'accès aux items d'un langage de représentation visuel \mathcal{L} est l'inefficacité maximum qui peut exister dans une de ses expressions

$$\mathcal{I}(\mathcal{L}) = \max_{e \in \mathcal{L}} \mathcal{I}(e)$$

Exemple 1 *Un exemple de représentation visuelle 100% inefficace pour l'accès à l'information, est un texte écrit dans lequel les mots sont les items et leur connexion est donnée par leur ordre dans le texte. Si nous considérons que l'entrée est le premier mot alors la plus grande distance a la taille du texte dans le pire cas où nous voulons accéder au dernier mot du texte depuis l'entrée.*

Ainsi si le langage contient 35 000 mots (comme le français par exemple) alors l'expression la plus inefficace sera une expression dans laquelle on a écrit les 35 000 mots consécutivement (éventuellement dans un ordre aléatoire), la distance maximale sera donc de 34 999 dans le cas où l'entrée est le premier mot (c'est la distance entre le premier et le dernier mot).

Dans cette section, nous proposons d'énumérer un ensemble de postulats qu'un langage visuel efficace et bien adapté au fonctionnement humain devrait satisfaire. Nous donnons d'abord 4 postulats qui s'expriment facilement dans les notations que nous avons présentées.

- **Accessibilité** $\forall e \in \mathcal{L}, \forall i \in items(e), \exists k \geq 0, \exists x \in entrySet(e)$ t.q. $path_k(x, i) \in e$.

2. Nous utilisons la définition classique d'un chemin dans un graphe, ici les arcs sont les connexions et les sommets sont les items, avec la convention qu'un chemin de longueur nulle (appelé chemin vide) existe de n'importe quel item à lui-même.

- **Navigation** : $\forall e \in \mathcal{L}, \forall i, j \in \text{items}(e)$ si $\exists k \geq 0, \exists x \in \text{entrySet}(e)$ avec $\text{path}_k(x, i) \in e$ alors $\exists k' \geq 0$ t.q. $\text{path}_{k'}(i, j) \in e$.
- **Efficacité d'accès** : $\mathcal{I}(\mathcal{L}) < 100\%$.
- **Entrée** : $\forall e \in \mathcal{L}, |\text{entrySet}(e)| = 1$.

En d'autres termes, l'*Accessibilité* impose que toute information exprimée devrait être accessible à l'utilisateur. *Navigation* exprime le fait que depuis n'importe quelle information accessible, l'utilisateur peut accéder à n'importe quelle information exprimée. L'*Efficacité d'accès* impose que n'importe quelle information doit être facilement accessible, c.-à-d., on ne doit pas avoir à lire tout le document afin de la trouver. *Entrée* exige que toute expression a un point d'entrée unique.

Les propriétés suivantes sont des conséquences de ces 4 principes :

Proposition 1 *Si un langage \mathcal{L} satisfait Accessibilité et Entrée alors le graphe des connexions de toute expression $e \in \mathcal{L}$ est connexe.*

Navigation et *Accessibilité* impliquent que de n'importe quelle position il doit toujours être possible de revenir à une position déjà examinée précédemment :

Proposition 2 *Si un langage \mathcal{L} satisfait Navigation et Accessibilité alors la relation de connexion est symétrique.*

Notons que la distance de n'importe quel item au point d'entrée (quand l'entrée est unique) joue un rôle important par rapport à l'*Efficacité d'accès*.

Proposition 3 *Soit \mathcal{L} un langage satisfaisant Entrée et Accessibilité et Navigation, $\forall e \in \mathcal{L}$, notons x_e l'unique élément de $\text{entrySet}(e)$, si x_e est le centre de gravité de $\text{items}(e)$ (par rapport à la distance de x_e aux items) alors $\mathcal{I}(e) < 100\%$.*

Les postulats suivants sont écrits en langage naturel puisqu'ils nécessiteraient d'avantage de notations pour les notions impliquées, l'écriture formelle de ces postulats sera l'objet d'une étude future plus poussée.

- **Correspondances de similarités** : Les similarités des positions/formes/couleurs doivent avoir des correspondances en termes de proximité de certains attributs des éléments représentés.
- **Significativité** : les positions/formes/couleurs des items ont une signification claire (le centre est important, être à gauche et à droite peut se rapporter à certaine contrainte de priorité, etc.)
- **3D** : le langage doit employer les 3 dimensions (environnement naturel des êtres humains), par conséquent la représentation en 2 dimensions devrait être combinée avec une possibilité de zoom avant/arrière.
- **Brièveté, puissance suggestive et clarté des symboles** : chaque symbole doit être court et simple. « Court » signifie moins de sept éléments, selon les propriétés de la mémoire temporaire de travail [14]. La puissance suggestive pourrait être mesurée en employant la théorie du « double codage » (dual-coding theory [17]).

- *Limitation de la surcharge cognitive* : le nombre d'éléments présentés simultanément à l'utilisateur doit être limité (selon la théorie de Sweller [28]).
- *Facile à écrire* : des outils conviviaux doivent être disponibles,
- *Traduisible* : toute expression visuelle valide doit être traduisible dans un formalisme logique et réciproquement.
- *Cohérence vérifiable* : il doit y avoir des règles pour vérifier si n'importe quelle expression a au moins une traduction valide.

Notez que tous ces axiomes sont liés à d'autres principes importants et souhaitables. En effet, nous pourrions définir un principe de *Voisinage* disant que : Toutes les informations ayant des propriétés communes doivent être proches en distance. Ce postulat implique le postulat *Correspondance de similarités* relativement à la position.

Le postulat *Brièveté, puissance suggestive et clarté des symboles* implique que le langage est *Facile à lire* il signifie que les symboles sont compréhensibles sans apprentissage préalable et que les expressions seront faciles à comprendre et mémoriser [17]. Le postulat *Significativité* peut impliquer que le centre de la représentation a une importance. *Traduisible* implique qu'une expression a une signification univoque et permet d'employer des mécanismes d'inférence.

Notre but est d'établir un langage visuel qui satisfait le plus grand nombre de ces postulats. Nous commençons par rappeler la définition du langage visuel MOT puis nous introduisons le nouveau langage VTL, nous terminons en montrant les principes qu'il satisfait.

3 Le langage visuel « MOT »

3.1 Description

Nous avons basé la construction de notre langage VTL sur l'approche MOT « Modélisation par objets typés » proposée par Paquette [19]. Cette méthode de représentation des connaissances est dédiée à des formateurs qui doivent définir des systèmes d'apprentissage ou des systèmes d'aide à la réalisation de tâches. Cette approche est basée sur un formalisme graphique.

Selon son auteur, les buts de MOT sont :

- simplicité d'usage pour des personnes non aguerries aux techniques de modélisation des connaissances,
- représentations adaptées à une grande variété de situations et de domaines,
- vue claire des relations entre connaissances, permettant de couvrir tout un domaine sémantique.

Les deux principes fondamentaux de MOT sont :

1. Toute connaissance peut être représentée par un élément dont la forme dépend d'un des trois types de connaissance : déclarative, action ou stratégique. La bordure de l'élément est soit en trait plein pour un élément abstrait soit en traits discontinus pour un élément factuel :

Type de connaissance	Abstraite	Factuelle
Déclarative	Concept	Exemple
Action	Procédure	Trace
Stratégique	Principe	Énoncé

2. Il y a 6 types de liens entre connaissances : instanciation (I), spécialisation (S), composition (C), précédence (P), Intrant/Produit (I/P) et régulation (R). Tout lien qui ne relève pas de ces 6 types peut être représenté en utilisant un attribut interne.

Voici une description plus précise des 6 types de liens dans MOT :

- I : toute connaissance abstraite (concept, procédure, principe) peut être liée à une de ses instances (connaissance factuelle)
- S : tout concept/procédure/principe abstrait peut être organisé en hiérarchies
- C : tout attribut s'il est assez complexe peut être externalisé en un nouveau schéma relié au premier par un lien de composition
- P : Toute procédure peut être décomposé en sous-procédures reliées entre elles par des liens de précédence
- I/P : La notion de procédure peut admettre des entrées/sorties (Intrant/Produit) qui sont des instances ou des concepts abstraits selon le niveau de généralité de la procédure
- R : Des connaissances peuvent régir ou contrôler d'autres connaissances grâce au lien de régulation.

Les exemples présentés sur les figures 1 et 2 sont issus de wikipedia [30], ils utilisent MOT pour représenter la procédure « Gérer les déchets », (cet exemple est présenté en anglais afin de mieux pouvoir le comparer avec son équivalent en VTL réalisé dans cette langue, notons que les liens I/P sont notés I/O (pour input/output)).

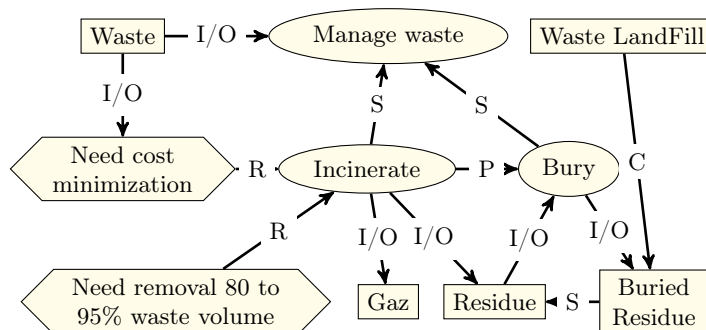


FIGURE 1 – « Manage Waste » dans le formalisme MOT

Dans le travail original de Paquette, il impose des contraintes d'intégrité sur les liens. À savoir, un lien ne peut pas exister par lui-même, il doit avoir

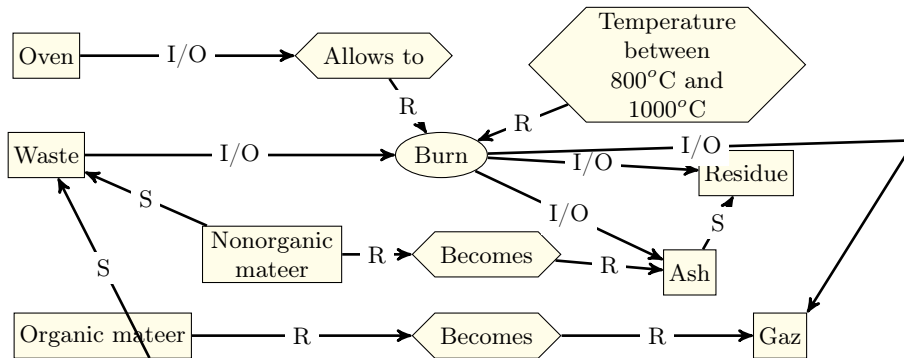


FIGURE 2 – Sous-modèle « Incinerate » dans le formalisme MOT

une origine et une destination qui doivent être des connaissances abstraites ou factuelles. Un lien peut connecter une connaissance à elle-même (à la fois origine et destination). Une connaissance peut être liée à aucune, une, ou plusieurs connaissances. Entre deux types de connaissance, les seuls liens valides considérés sont donnés dans le Tableau 1. S'il y a un lien entre deux connaissances alors il est unique avec un type unique. S'il y a plusieurs destinations pour un lien S, I/P ou I alors ces destinations doivent avoir le même type (ce qui n'est pas exigé pour les autres liens).

Tableau 1 – Liens valides entre connaissances

De	Connaissance Abstraite			Connaissance Factuelle		
	Concept	Procédure	Principe	Exemple	Trace	Énoncé
Concept	S, I, C	I/P	R	I, C	I/P	R
Procédure	I/P	C, S, P, I	C, P, R	I/P	I, C, P	R, P, C
Principe	R	C, R, P	C, S, P, R, I	R	C, R, P	I, C, P
Exemple	C	I/P	R	C	I/P	R
Trace	I/P	P, C	P, R, C	I/P	C, P	C, P, R
Énoncé	R	R	R	R	C,R,P	C,R,P

De plus, Paquette [20] a imposé que les relations de Spécialisation (S), Composition (c) et Précédence (P) soient des ordres strict partiel (irréflexifs, transitifs, asymétriques et non totaux)³ et que les liens Intranç/Produits (I/P), Régulation (R) et Instanciation (I) ne soient pas transitifs.

L'approche MOT est intéressante parce qu'elle a introduit la notion d'éléments et de relations typés, nous allons toutefois exprimer quelques critiques qui justifient la tentative de définir un nouveau langage.

3. Par soucis de simplicité, les liens transitifs ne sont pas exprimés dans les schémas.

3.2 Critiques

Un des objectifs de MOT était d'être facile à apprendre et à comprendre. À notre avis, ce but n'est pas atteint. Les lettres employées pour différencier les différents types de liens ne sont pas très explicites, il serait plus faciles d'employer des formes, des couleurs, des mots, des symboles ou des émoticônes... Les formes standard de MOT n'ont pas de signification intrinsèque, par exemple, les procédures ne sont pas modélisées en prenant en compte la notion temporelle/causale forte qui les caractérisent (elle pourrait être capturée par une roue dentée ou une flèche...). Par ailleurs, les principes n'ont pas de définition très cadrée, par conséquent ils sont souvent représentés par des phrases complètes, ce qui va à l'encontre de l'idée d'employer des modèles visuels privilégiant la simplicité et la clarté. Par conséquent le postulat *Brièveté, puissance suggestive et clarté des symboles* n'est pas satisfait par MOT.

Des liens de Régulation semblent être employés d'une manière inexacte dans l'exemple fourni par l'auteur : un lien de Régulation est employé pour exprimer la manière (par exemple « burn with an owen » dans la figure 2) ou pour exprimer des liens de priorité entre les concepts, cette utilisation n'est ni claire ni univoque violant les postulats de *Correspondance de similarités* et *Significativité*.

D'autre part, les diagrammes complexes peuvent être difficiles à appréhender, violant le principe d'*Efficacité d'accès* et de *Limitation de la surcharge cognitive*. La méthode MOT n'offre pas la possibilité d'effectuer des zoom avant/arrière afin de faire une projection selon certaines relations d'intérêt ou certains attributs précis : la géographie, la causalité, la spécificité, etc. Par exemple, les relations de Composition et d'Instances sont des relations qui changent le niveau de détails, par conséquent, elles pourraient être associées à certaines opérations de zoom avant ou arrière. Ceci viole le postulat *3D*.

4 Notre projet : VTL

Comme dans MOT, nous proposons d'utiliser trois types d'éléments : actions, entités et conditions. Notre amélioration porte plutôt sur les relations entre ces différents éléments.

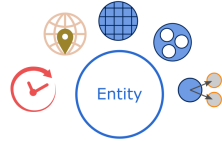
4.1 Les 3 types d'éléments

Les trois types d'éléments peuvent être génériques (bordure bleue et fond blanc) ou spécifiques (bordure orange et fond gris). Chaque élément générique peut être accompagné du symbole *Instances*.



La fonctionnalité *Instances* décrit des exemples particulier d'une entité générique. Pour accéder aux entités génériques auquel l'entité spécifique courante se réfère on utilise un lien hypertexte. Par exemple, « four n°ref F118 » est une instance de « four ».

4.1.1 Entités



Chaque entité (générique ou spécifique) est représentée

par un cercle, avec un intitulé et une icône (optionnelle) à l'intérieur. Cette forme est associée à cinq symboles. Les cinq symboles dessinés ci-dessus, en commençant de la droite vers la gauche représentent respectivement les caractéristiques *Instances* ou *Instance de* (selon que l'entité est générique ou spécifique), *Composé de*, *Propriétés*, *Spatialisation* et *Temps*. Cela permet d'associer les entités à certaines spécificités et permet en outre de naviguer par projection sur une caractéristique spécifique.



La fonctionnalité *Composé de* peut décrire un ensemble d'entités qui composent l'entité principale. Par exemple, une voiture est composée d'un volant, de quatre roues, d'un moteur, etc.



La fonctionnalité *Propriétés* peut décrire des caractéristiques spécifiques de l'entité. Par exemple, un « déchet » peut être « macroscopique », ou « jaune » ou « alimentaire ».

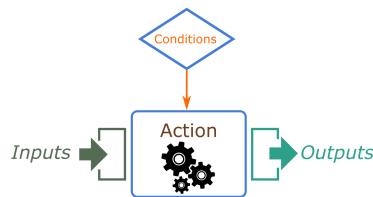


La fonctionnalité *Spatialisation* est généralement utilisée pour localiser l'entité dans le monde, dans une pièce, etc. Par exemple, il est possible de spécifier les coordonnées GPS, les volumes et les zones, ainsi que les positions relatives (au-dessous/ au-dessus /à gauche /à droite d'une autre entité).



La caractéristique *Temps* quant à elle permet de localiser l'entité dans un temps universel ou relatif (c'est-à-dire, par rapport à d'autres entités). Par exemple, une voiture peut avoir une date de naissance, et une durée de vie moyenne.

4.1.2 Actions



Les actions sont symbolisées par un rec-

tangle bleu avec une étiquette de couleur brune associée au dessin d'un engrenage. Les actions sont liées à des entrées, des sorties et des conditions.



Les entrées sont les entités nécessaires pour exécuter l'action, ou les

entités qui sont intéressantes à mentionner pour que l'action ait lieu. Par exemple « déchets » et « four » sont des entrées pour l'action « brûler des déchets ».



Les sorties sont des entités résultant d'une action ou qui ont un certain intérêt à être mentionnées après une action. Par exemple, l'action « brûler des déchets » est liée à un ensemble de sortie telles que « résidus », « gaz » et « four ».

Les conditions sont décrites dans la section suivante.

4.1.3 Conditions

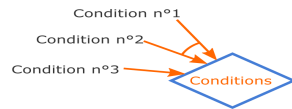


Les conditions sont représentées par des losanges bleus avec un libellé orange. Certaines actions nécessitent des conditions pour se réaliser.

Dans notre exemple de la figure 4 : le four doit être en état de marche.

Plus généralement, les conditions peuvent s'exprimer sur toutes les entités, même si elles ne sont pas liées à des actions. Plusieurs conditions peuvent être connectées par des opérateurs logiques et / ou. Nous utilisons la convention des *arbres et-ou* pour représenter des combinaisons de conditions.

Par exemple, la condition définie par ((condition n°1) et (condition n°2)) ou (condition n°3) est représentée par :



4.2 Relations

Dans VTL, les relations entre entités, actions et conditions sont représentées au moyen des 5 caractéristiques prédéfinies associées aux entités, ou par les 3 caractéristiques associées aux actions ou encore par une combinaison de conditions. Cela nous permet de mettre en avant les relations suivantes.

- Relations d'entités à entités : comme par exemple *Composé de*, *Instances*, *Forme générique de*, *Se déroule avant*, *Se déroule après*, *Au nord de*, etc.
- Relations entre entités et actions : via les entités entrée/sortie propres aux actions.
- Relations entre conditions et entités : certaines conditions peuvent être exprimées sur les entités. Elles peuvent être considérées comme des filtres sur ces entités.
- Relations entre conditions et actions : ces relations sont des contraintes sur l'exécution possible d'une action.
- Relations entre conditions et conditions : les relations entre les conditions sont symbolisées par la convention et-ou (comme on l'a vu dans la section 4.1.3)

4.3 Navigation

VTL est associé à un outil de navigation. En effet, afin de permettre une représentation plus claire, au lieu d’avoir un vaste graphique d’entités, il est possible de ne représenter qu’un sujet principal sans trop de détails. Ensuite, le navigateur peut naviguer par zoom avant/arrière sur une caractéristique précise pour n’obtenir que les détails qui l’intéressent. En cliquant sur une entité, l’utilisateur obtient une nouvelle vue dont l’entité occupe le centre. La navigation est ainsi un moyen d’explorer certaines caractéristiques précises. La navigation est illustrée par la Figure 3 disponible en ligne [8].

4.4 Exemple

La figure 4 décrit une représentation en VTL de la gestion des déchets décrite en MOT par les figures 1 et 2. Cette figure représente l’action générique de brûler les déchets et une instance de cette action peut être examinée en cliquant sur « Brûler-278 » à partir de l’attribut *Instances* de l’action principale « Brûler des déchets » (Figure 5).

5 Propriétés de VTL

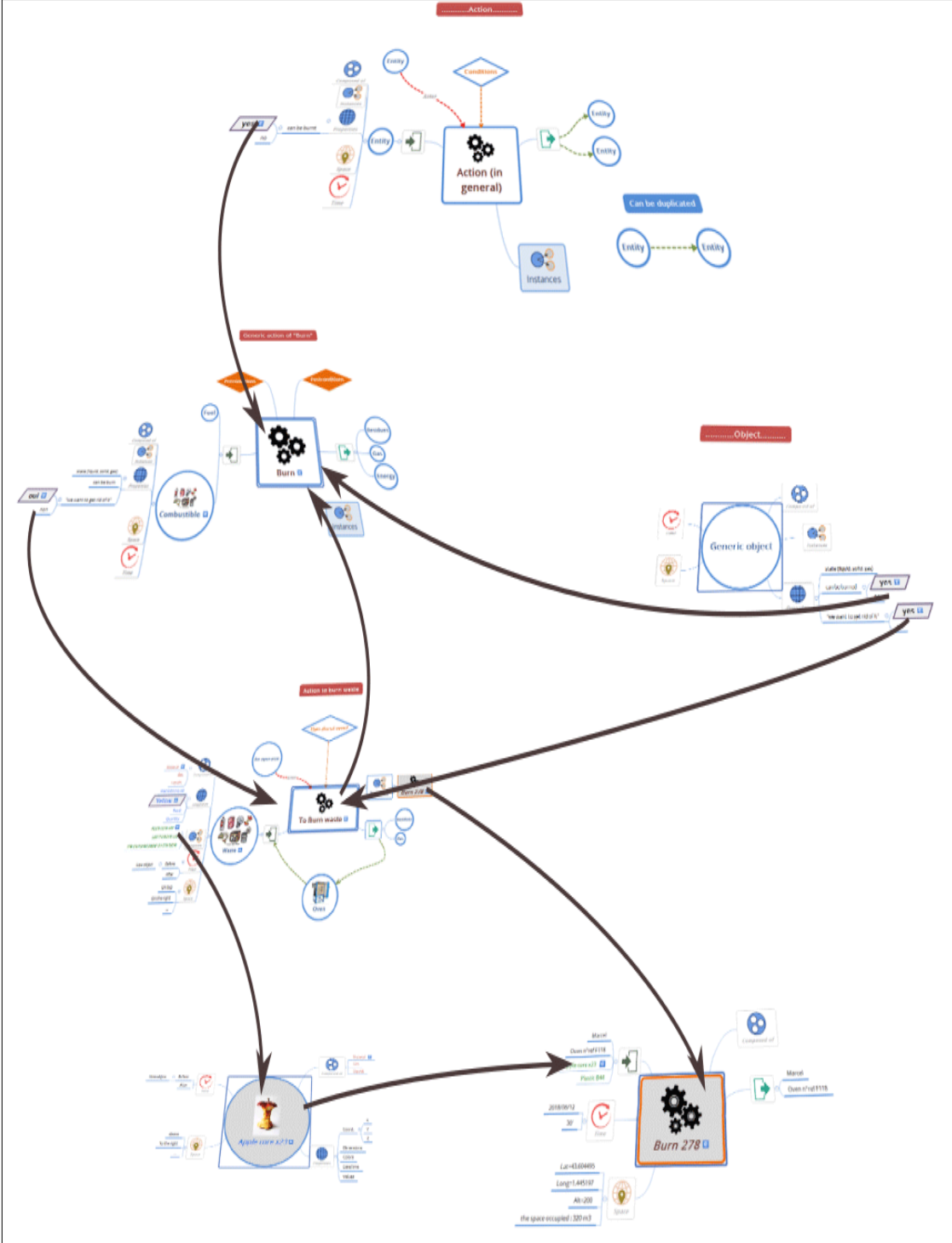
5.1 Propriétés sémiotiques

Comme mentionné dans l’introduction, nous nous appuyons sur les travaux et recherche liés à la sémiologie. Ceux de *Jacques Bertin* [2], cartographe français, a ainsi mis en avant dans son ouvrage majeur « Sémiologie graphique » un ensemble de 8 « variables visuelles ». Ces travaux étant plutôt dédiés à la représentation des données, nous ne tiendrons compte que de quelques-unes de ces variables visuelles : comme le changement de couleur pour signifier un changement de catégories (« entité générique » ou « instance » pour nous), de formes (« entités » / « actions » / « conditions » dans notre cas) et de taille de ces dernières. En l’état actuel de nos travaux, ce dernier élément n’est pas complètement implémenté. Nous excluons les autres variables visuelles (x, y⁴, grain, valeur d’une couleur, orientation) principalement utilisées dans les cartes géographiques.

La théorie de la *Gestalt* (« théorie de la forme ») est également une source d’inspiration pour nous. Elle fait référence à un ensemble de lois que la perception visuelle humaine met en œuvre de façon automatique. La loi de proximité par exemple, qui crée implicitement une notion de groupement d’entités proches les unes des autres, sera abondamment utilisée. De même, le principe de continuité, permettra dans VTL de guider l’œil du lecteur pour l’amener « naturellement » au symbole visuel suivant, sans rupture (tel que celui généré par un saut de page par exemple). L’utilisation de lien hypertexte dans la version actuelle du prototype de VTL est cependant en contradiction avec ce principe.

4. La position des éléments dans la représentation VTL est cependant prise en compte mais pas de manière absolue.

FIGURE 3 – Navigation en VTL



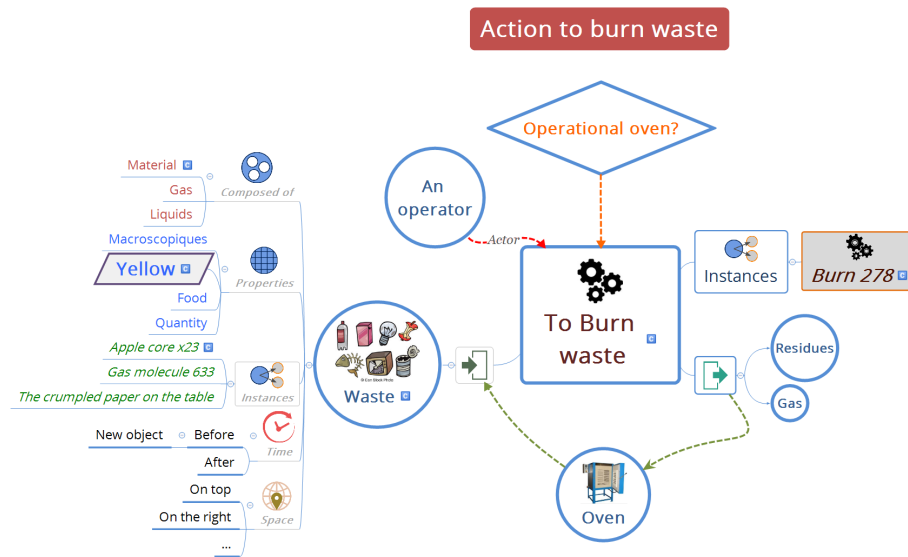


FIGURE 4 – L'action « To Burn Waste » en VTL

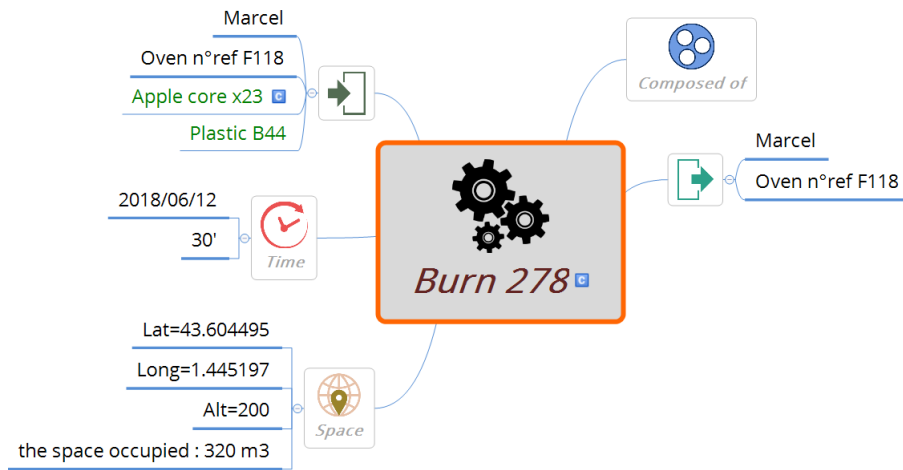


FIGURE 5 – Instance de l'action Burn en VTL

Nous envisageons de corriger cet inconvénient dans la version suivante de VTL et d'ajouter le principe de *Continuité* à nos axiomes.

Comment avons-nous choisi les symboles visuels de VTL ?

Les *entités* et les *attributs* associés sont tous dans des *cercles*. Cette forme est très souvent utilisée pour symboliser quelque chose d'indéfini, de non précisé. On la retrouve dans de nombreux logiciels pour représenter un objet, une personne, une abstraction, etc.

Les symboles choisis pour les attributs sont décrits ci-dessous.



La forme *Composé de* montre un cercle qui en englobe d'autres. On visualise implicitement une structure arborescente qui correspond bien au « tout » composé de « parties » ;



Le symbole *Propriétés* affiche quant à lui une grille, un filtre : parmi toutes les propriétés possibles de l'entité en question, seules quelques-unes sont utilisées, filtrées, sélectionnées ;



Le symbole *Instance* utilise la forme des entités, dont l'une est la "source" (en bleu) et les deux autres "instanciées" (en gris). Le changement de couleur est nécessaire afin d'indiquer qu'il ne s'agit pas de la même nature d'objets ;



Spatialisation fait référence à une mappemonde et à marqueur proche de celui utilisé dans Google Maps, deux symboles qui ne laissent aucune ambiguïté sur la localisation dans l'espace de l'entité ;



Le symbole *Temps* combine également 2 références visuelles : les aiguilles d'une horloge (référence au temps) au centre d'un cadran fléché pour signifier la notion de durée.



Le symbole associé à une *Action* comporte un engrenage qui évoque un mécanisme en mouvement. En accord avec les variables visuelles de J. Bertin, nous utilisons une forme différente de celles des entités, le carré est choisi car c'est également cette forme que l'on retrouve dans la cartographie de processus.



Les entrées et sorties sont représentées par des portes.



La forme des *Conditions* reprend celle utilisée dans les logigrammes pour tester une variable et aiguiller le déroulement du procédé.

Conscient que ces choix sont subjectifs nous envisageons de développer un protocole expérimental de validation par des utilisateurs du langage VTL et

plus précisément des symboles utilisés. Au-delà de cette validation expérimentale, il serait intéressant d'étudier la possibilité de mettre en place des principes formels sémiotiques (comme le principe de continuité). Un autre principe sémiotique pourrait être mis en place pour les contraires : il faudrait imposer une cohérence en couleur/forme/taille entre contraire. Ici, par exemple, les couleurs des portes Entrée et Sortie sont différentes mais les couleurs pas significative de la fonction contraire. Cependant ce n'est pas encore l'objet de notre présente contribution. Notons que cette piste a été explorée pour le langage VCM (Visualization of Concepts in Medicine [12]). Le langage VCM propose une liste de symboles de base représentant des concepts médicaux que l'utilisateur peut combiner pour former de nouveaux symboles. Cela permet d'accompagner les textes médicaux avec ces images, chaque image est facile à comprendre à partir de sa décomposition en symboles de base. L'utilisation de VCM satisfait ainsi notre principe de *Brièveté, puissance suggestive et clarté des symboles*. La démonstration du bien fondé du langage VCM donnée dans [12] se base sur l'association d'une ontologie OWL aux symboles dont la traduction en DL a permis de prouver la cohérence hiérarchique des icônes.

5.2 Propriétés théoriques

Il est possible d'imposer dans VTL que tout schéma n'ait qu'une seule entrée qui serait le centre d'un graphe, ce qui impliquerait que les principes *Accessibilité, Navigation, Efficacité de l'accès* et *Entrée* seraient respectés. Dans VTL, les symboles que nous proposons sont associés à des mots courts et nous recommandons d'ajouter des images avec ces éléments. Par conséquent, nous incitons à utiliser le « double codage » qui implique la satisfaction du principe de *Brièveté, puissance suggestive et clarté des symboles*. Le double codage introduit par [18] consiste à

Par construction, les positions/formes/couleurs des trois types d'éléments ont été choisis dans VTL afin d'avoir une interprétation claire, ce qui signifie que le principe de *Correspondance de similarité* est respecté pour les formes et les couleurs, ainsi que le principe de *Significativité*.

Concernant les positions relatives des objets, leur *Significativité* n'est pour le moment imposé que pour les actions (où les entrées sont sur la gauche tandis que les sorties sont à droite, et les conditions sont au-dessus).

La navigation dans VTL est faite de manière à donner la possibilité de zoomer en avant/en arrière (voir Figure 3), ce qui permet de satisfaire le principe *3D*. Le principe de *Limitation de la surcharge cognitive* est un postulat qui devra être imposé aux auteurs de VTL. Cette condition peut être imposée sans perte d'information en utilisant les fonctionnalités natives de VTL (navigation et zoom avant/arrière).

Pour illustrer notre propos, nous avons utilisé le logiciel XMind qui nous a permis de créer très facilement un exemple complet. D'où le postulat sur la *Facilité d'écriture* même si un outil dédié serait plus adapté. Concernant le postulat *Traduisible*, l'idée d'utiliser un langage typé nous permet d'imposer des restrictions sur les types d'éléments autorisés et également sur leurs connexions.

Une traduction automatique de toute expression sera le sujet d'une prochaine étude. Le principe de *Cohérence vérifiable* n'est pas encore en œuvre en VTL, il sera possible lorsque nous disposerons d'un outil de traduction formelle.

6 Conclusion et travaux connexes

Nous avons proposé plusieurs principes visant à régir les langages de représentation visuelle des connaissances et avons défini un nouveau langage de représentation typé VTL compatible avec ces principes. En VTL l'information est accessible par navigation à l'intérieur d'une structure arborescente. En effet, à un instant donné, l'accès à tous les détails d'un sujet donné est superflu pour l'utilisateur. Au contraire, il est nécessaire de limiter les détails exposés d'emblée afin de ne pas lui imposer une charge mentale trop importante. Ainsi, la navigation dans VTL permet à l'utilisateur d'accéder aux niveaux de détails voulus seulement pour les informations qui l'intéressent spécifiquement.

Comme nous le signalons dans la précédente section, la validation sémiotique de notre langage par des tests avec des utilisateurs n'est pas encore envisagée. Nous n'en sommes pas au stade de la définition d'un protocole expérimental qui pourrait étudier la facilité de passage dans un sens ou dans l'autre d'un texte écrit à sa représentation en VTL. Nous pourrions utiliser ce protocole pour comparer VTL avec d'autres langages visuels standards comme UML.

Le logiciel XMind a été utilisé pour créer des exemples et pour simuler la navigation, mais le développement d'une interface utilisateur graphique (GUI) spécifique à VTL est à l'étude. De plus, notre prochaine étape consistera à étudier la traduction de VTL dans un langage non visuel formel afin de proposer des inférences et des contrôles de cohérence.

Les notions d'héritage de propriétés sont traduites en VTL par navigation vers un niveau plus spécifique ou plus générique. L'héritage de propriétés n'est pas un concept nouveau inventé avec les langages objets puisque d'après Sowa [27], ça n'est qu'« un cas particulier des syllogismes d'Aristote dans lequel on spécifie les conditions d'héritage de propriétés d'un type vers un sous-type ». De même, Sowa souligne que la distinction entre instances et catégories est déjà présente dans la logique médiévale et se retrouve en bas du dessin de l'arbre de Porphyre, attribué à Pierre d'Espagne, et daté de 1239 (voir Figure 6). Ces notions de relations variées entre concepts associées à des relations d'héritage sont présentes dans les réseaux à héritage de structure (*structure inheritance networks* comme le langage KL-ONE (Knowledge Language One) [3] voir Figure 7. Un des points intéressants de l'arbre de Porphyre comme d'UML est l'existence d'attributs appelés *Differentiae* qui permettent de séparer un type d'un sous-type.

Une autre direction de travail consisterait à étudier comment VTL permet d'englober les liens qui ne sont pas traités par MOT, à savoir les relations RCC8 [22] ou les intervalles d'Allen, ou d'autres relations entre concepts (on pourrait se référer par exemple à la typologie des mots-liens écrit par Christian Barette [1]).

FIGURE 6 – L'arbre de Porphyre selon Pierre d'Espagne (1239) [27]

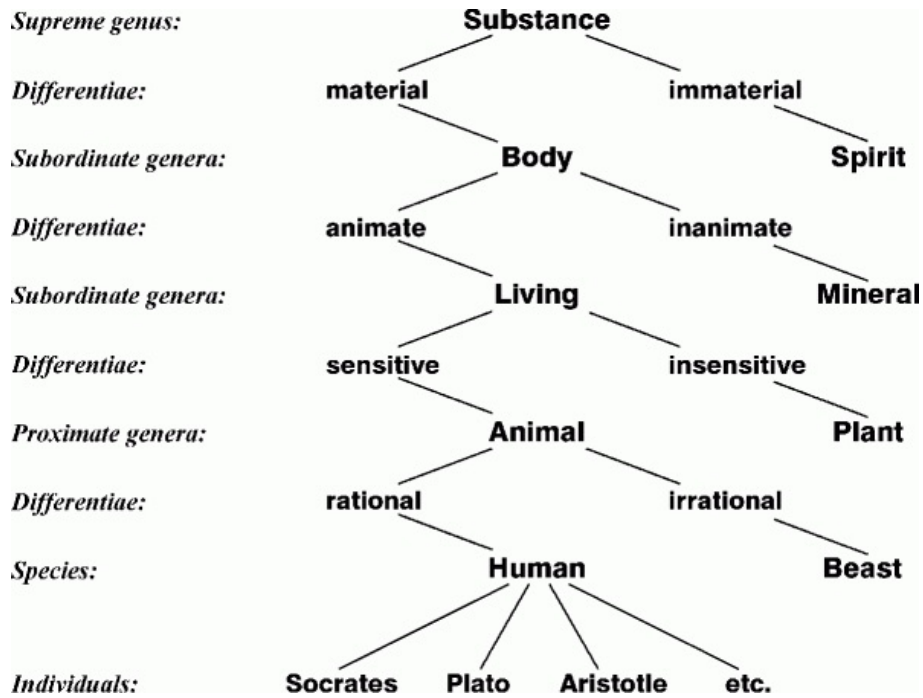
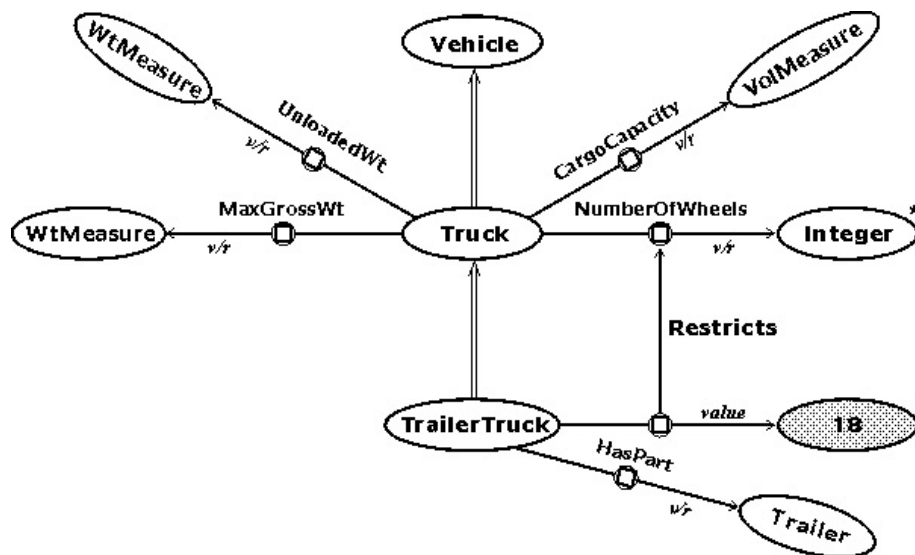


FIGURE 7 – Représentation en KL-ONE d'un camion [27]



Dans l'idée de raisonner visuellement, nous envisageons la possibilité de transformer le dessin par inférence, c'est un peu réalisé dans le langage VCM [12] puisqu'il permet de créer des combinaisons de symboles à partir de symboles. Cependant une autre façon de réaliser ce raisonnement est d'écrire des règles sémantiques d'insertion ou de retrait d'éléments graphiques, comme cela a été proposé en 1896 par Charles S Pierce avec les graphes existentiels (*existential graphs*) [23]. Les graphes existentiels sont des représentations dans lesquels on peut entourer des lettres (pour la négation) ou les juxtaposer (pour la conjonction), ce qui permet de visualiser une formule logique, avec des règles disant que deux entourages d'un même élément peuvent être supprimés (double négation). Bien que la représentation proposée ne soit pas très intuitive pour la perception visuelle, la gestion sémantique d'énoncés logique faite par Pierce pourrait nous inspirer afin de mettre en place des règles de modifications (ou d'inférence) en VTL.

Remerciements

Les auteurs remercient les rapporteurs anonymes du comité de programme de la conférence IC'2018 pour leurs compétences et leur exigence. Leurs remarques pertinentes ont permis de faire progresser cet article de façon substantielle.

Références

- [1] C. Barette. L'analyse des relations. Technical report, Université de Sherbrooke, Québec, 2002.
- [2] Jacques Bertin. Sémiologie graphique : Les diagrammes-les réseaux-les cartes. *Gauthier-VillarsMouton & Cie*, 1973.
- [3] Ronald J Brachman, Deborah L McGuinness, Peter F Patel-Schneider, Lori Alperin Resnick, and Alexander Borgida. Living with classic : When and how to use a kl-one-like language. *Principles of semantic networks*, pages 401–456, 1991.
- [4] T. Buzan. *Use your head*. BBC Books, London, 1974.
- [5] Melvin E Dickover, Clement L McGowan, and Douglas T Ross. Software design using : Sadt. In *Proceedings of the 1977 annual conference*, pages 125–133. ACM, 1977.
- [6] Desmond F D'Souza and Alan Cameron Wills. *Objects, components, and frameworks with UML : the catalysis approach*, volume 1. Addison-Wesley Reading, 1998.
- [7] Florence Dupin De Saint Cyr and Denis Parade. Knowledge Representation in a Visual Typed Language : from Principles to Practice. In *International Conference on Information Visualisation (IV)*, KV : Knowledge Visualization and Visual Thinking. IEEE, 2018.

- [8] Florence Dupin de Saint Cyr and Denis Parade. Maquette interactive VTL sur l'exemple Burn Waste, 2018.
- [9] Michel Grundstein. Gameth : un cadre directeur pour repérer les connaissances cruciales pour l'entreprise. *Lamsade Université Paris-Dauphine*, 2002.
- [10] Michel Grundstein and Camille Rosenthal-Sabroux. Gameth[®], a decision support approach to identify and locate potential crucial knowledge. In *Proc. 5th Europ. Conf. on Knowledge Management*, pages 391–402, 2004.
- [11] ISO. 5807 : 1985 information processing-documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts. *Geneva*, 1985.
- [12] Jean-Baptiste Lamy, Lina F. Soualmia, Gaëtan Kerdelhué, Alain Venot, and Catherine Duclos. Validating the semantics of a medical iconic language using ontological reasoning. *Journal of Biomedical Informatics*, 46(1) :56 – 67, 2013.
- [13] David A Marca and Clement L McGowan. *SADT : structured analysis and design technique*. McGraw-Hill, Inc., 1987.
- [14] George A. Miller. The magical number seven, plus or minus two : Some limits on our capacity for processing information. *Psychological Review*, 63 :81–97, 1956.
- [15] Otto Neurath. International picture language. the first rules of isotype. *London : K. Paul, Trench, Trubner & Co*, 1936.
- [16] Joseph D Novak and Alberto J Cañas. The theory underlying concept maps and how to construct and use them. Technical report, Institute for Human and Machine Cognition, 2008.
- [17] A Paivio. Mental imagery in associative learning and memory. *Psychological Review*, 76 :241-263, 1969.
- [18] Allan Paivio. *Mental representations : A dual coding approach*. Oxford University Press, 1990.
- [19] Gilbert Paquette. Visual knowledge and competency modeling-from informal learning models to semantic web ontologies. *Hershey, PA : IGI Global*, 2010.
- [20] Gilbert Paquette, Michel Léonard, Karin Lundgren-Cayrol, Stefan Mihaila, and Denis Gareau. Learning design based on graphical knowledge-modelling. *J. Educational Technology & Society*, 9(1), 2006.
- [21] Steve Pepper. The TAO of topic maps. https://www.researchgate.net/publication/225070304_The_TAO_of_topic_maps, pages 1–21, 01 2000.
- [22] David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. In *Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 165–176, 1992.

- [23] Don D Roberts. *The existential graphs of Charles S. Peirce*, volume 27. Walter de Gruyter, 1973.
- [24] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William E Lorensen. *Object-oriented modeling and design*, volume 199, 1. Prentice-hall Englewood Cliffs, NJ, 1991.
- [25] James Rumbaugh, Grady Booch, and Ivar Jacobson. *The unified modeling language reference manual*. Addison Wesley, 2017.
- [26] J.F. Sowa. *Conceptual structures—Information processing in mind and machine*. Addison-Wesley, Reading, MA, 1984.
- [27] John F Sowa. Semantic networks. *Encyclopedia of Cognitive Science*, 2006.
- [28] J. Sweller. Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4) :295–312, 1994.
- [29] James Teboul and Philippe Damier. *Neuroleadership*. Odile Jacob, 2017.
- [30] Wikipédia. Modélisation par objets typés, 2017.