Patrick PEREZ

Laboratoire d'Informatique Appliquée
à l'Architecture
Ecole d'Architecture de Toulouse

L'élément d'architecture usage et définition dans les systèmes constructifs et langages objets

Résumé :

Cet article est une tentative de présentation du dessin d'architecture, et plus précisement de la notion d'élément d'architecture dans les systèmes constructifs, vus sous l'angle d'un langage du projet, et d'un projet de langage, lors de la conception. Nous nous proposons d'analyser ce langage dans ses aspects syntaxiques et sémantiques, et dans les différents modes opératoires qui l'articulent. Cela, pour une meilleure représentation et manipulation des connaissances en C.A.A.O, mettant l'accent sur le problème de la capacité d'expression d'un "savoir-faire".

Abstract :

This article is an attempt to present the architectural design, and more precisely the notion of the architectural element in the constructive systems, considered as a language of the project, and as a project of language at the time of conception. We purpose to analyse this language in its syntactic and semantic aspects and in the different operative modes which joint it, this for a better representation and manipulation of knowledges in C.A.A.D, pointing out the problem of the expressing capability of a "savoir-faire". (or of an ability).

1. Introduction.

" Nous n'usons pas des choses mais de leurs images".
Senancour.

Le concepteur s'exprime au moyen d'un outil fonctionnel que nous nommons dessin. Polymorphe et polysémique, le croquis, l'épure, le rendu, nous apparaissent comme la forme écrite du langage de conception architecturale.

Le dessin est à la fois expression d'un but et quête.

Il fixe et précise un processus de validation, de jugement, et de remémoration. Il est aussi trace et désir (Songeons d'ailleurs à l'orthographe double et double acception, que dessin/dessein possédait jusqu'au 17eme sciècle).

Il joue donc à la fois, le rôle de message par une combinatoire de symboles où des signifiants précis s'approprient un réel existant ou en devenir, et un rôle de validation de la pensée dans le va et vient incessant qu'il stimule chez l'Architecte pour la quête de la 'bonne et belle réponse' à un problème donné.

Raccourci et véritable substitut de la réalité, le dessin d'architecture est un langage, dont nous nous proposons dans le présent article, de présenter certains aspects, ébauche théorique d'une possible implantation informatique au moyen des langages dits 'objets'.

2. L'élément d'Architecture.

Je regarde le petit dessin que j'ai sous les yeux et je pense : c'est une colonne. Le mot 'colonne', comme le dessin associé, évoquent en moi un objet réel que mes sens connaissent. Une colonne a un poids, une ou des couleurs, c'est un objet complexe fait de plusieurs parties. J'en connais constituées de diverses matières : de la pierre, de la terre cuite, du plâtre, du ciment, du bois même. Une colonne a aussi un diamètre spécifique qui lui permet de porter des objets plus ou moins lourds selon sa hauteur. Elle possède une histoire et je lui connais plusieurs apparences, détails et formes spécifiques...

En se sens, le petit dessin de la colonne est économie du réel, et si je change l'échelle de mon dessin, d'autres détails vont apparaitre, me précisant comment cette colonne doit être exécutée, le choix dans le possible des apparences, ses différentes parties, ses cotes... Le dessin de la colonne est donc un symbole, symbole indicatif quand je lis le plan (ou si un autre architecte lit le plan), il désigne alors le symbolisé, et substitutif quand je pense le plan dans l'acte de concevoir; le symbole idoine se substitue alors à la chose symbolisée, à l'objet réel (que peut-être je ne connais pas encore totalement).

L'élement d'architecture porte donc divers symboles associés aux différentes échelles du dessin, symboles où plus l'échelle est petite, plus le symbole devient de type associatif externe (la forme codique en vue de la construction), et paradoxalement, simultanément imitatif.

Mais l'élément d'architecture fait également appel à un passé partagé entre les différents lecteurs du plan ; la connaissance que nous avons de l'élément dans différentes situations de sa réalité concrète.

L'ensemble des différents éléments d'Architecture que je manipule dans un projet peut alors être considéré comme vocabulaire. Chaque symbole d'élément est constitué de plèmes absolus, les traits que

je dessine (plèmes absolus car chaque trait pris à part n'est pas porteur de sens en soit), et les associations de symboles formant nouveau symbole à signifié différent, redonnent aux symboles élémentaires statut de plèmes à leur tour, mais de plèmes relatifs.

On retrouve ici dans le dessin d'Architecture, la double discontinuité du langage naturel: la suite de plèmes absolus que constituent les lettres d'un mot dans le langage écrit, ou les phonèmes du langage parlé, et la suite de plèmes relatifs que constituent les mots d'une phrase.

3. Macro-éléments descriptifs et macro-éléments conceptuels.

Un élément d'Architecture peut donc être à son tour constitué d'éléments de base ayant une sémantique propre, et dont l'"addition sémantique" ne fournira pas le signifié général. Prenons l'exemple de la colonnade : il ne suffit pas de dire qu'une colonnade est une suite de colonnes pour avoir décrit la chose. Une colonnade fait appel à un ensemble de règles précises et subtiles portant sur les entraxes et le nombre, articulées par le type d'élément de base choisi (la colonne qui peut être dorique, toscane, composite etc...).

Pourtant le concepteur peut, pour une même échelle donnée, soit dessiner très correctement la colonnade symbolique par application de la règle 'faire une colonnade à partir d'une colonne donnée' (règle qui peut être d'interprétation très personnelle), le concepteur génère alors un macro-élément descriptif, soit 'gribouiller' une série de traits (des plèmes), créant ainsi un signifiant personnel.

Dans le second cas, le signifiant personnel peut très bien être symbole temporaire de symbole (le concepteur sait très exactement quel macro-élément il symbolise ainsi, mais il en remet le dessin définitif à plus tard pour des raisons d'efficacité conceptuelle), mais peut aussi signifier un macro-élément conceptuel. Il s'agit là d'un signifiant d'un type très particulier pouvant symboliser un objet virtuel (tel un tracé régulateur ou un gabari), ou un macro-élément descriptif dont l'exploration définitive est remise à plus tard.

Cette capacité de post-définition d'une sémantique totale et précise permet au concepteur de focaliser son esprit sur un ensemble de caractéristiques spécifiques et contingentes à un problème donné, signe et outil d'une méthodologie de réduction de la complexité, que nous pourrions nommer par analogie avec la théorie de la programmation, analyse descendante. Alors que les macro-éléments descriptifs, très utilisés en systémique, font plutot appel à une analyse ascendente.

Les macro-éléments ont une position de messages partiels, articulations et véritables phrases ou mélodies coupées de silences, dans le message général.

4. Micro-code et variation morphologique.

Le vocabulaire architectural possède un ensemble de lois de variations morphologiques applicables aux éléments de base, que nous avons définis à travers leurs symboles comme plèmie relative (pour une échelle donnée).

Un plème relatif référant alors un sème, par le rapport au réel qu'il entretient en tant qu'unité symbolisée distincte, nous parlerons maintenant de sèmie pour désigner l'ensemble des éléments architecturaux de base composant le vocabulaire terminal.

Lorsqu'un macro-élément descriptif se constitue, il fait non seulement appel à un certain nombre de règles de syntaxe, mais encore, chaque sème le composant est succeptible de faire appel à des lois

de variations modulant les plèmes relatifs de la sémie dans un contexte syntaxique précis. Comme si les plèmes se conjuguaient ou se déclinaient !

Le symbole peut alors se modifier, ou se voir adjoindre un nouvel élément d'articulation. Cet ensemble d'articulations va de la jambe de force, au joint, au tirant etc...

Les lois de variation, ponctuelles et portées par les éléments, sont autant de réponses à des problèmes physiques ou esthétiques de situation d'un élément dans un contexte spécifique. Si nous voulons faire un parallèle avec le langage naturel, considérons par exemple que le morphème lexical 'vert' peut, par application d'une loi de variation, se transformer en 'verte' ou en 'vertes', ou encore en 'vertement' ('e','es','ement' étant des morphèmes grammaticaux).

Un exemple simple d'un tel micro-code, est la loi de variation du diamètre de la colonne d'angle dans une colonnade. (le traitement du chapiteau d'angle suit également une loi de variation). Ces lois de variations sont, rappelons-le, intimement liées à l'usage d'un symbole dans un contexte précis. Elle sont comme 'portées' par des ensembles ou groupes de sèmes particuliers (filtrage et application catégorielle).

L'usage le plus représentatif des lois de modification du plème relatif dans le dessin d'architecture, est sans aucun doute le changement d'échelle. A un meme élément signifié peuvent correspondre plusieurs signifiants, mais chaque signifiant est propre à l'échelle à laquelle l'architecte travaille. D'autre part, de la forme du symbole à grande échelle, peut être extrait le symbole de la petite échelle, par recherche de la racine minimale signifiante (symboles gigognes !). A une échelle particulière, le projet est constitué d'une plèmie syntagmatique (les symboles des éléments ne peuvent se substituer avec une autre échelle sans détruire ou modifier la cohérence - et le 'sens' - du projet), mais globalement la plèmie est paradigmatique (ou mobile) sur l'ensemble du projet lors d'un changement d'échelle.

5. Aperçu syntaxique.

Après avoir étudié quelques aspects du vocabulaire architectural, observons comment celui-ci s'articule et se structure dans le projet. D'emblée nous devons distinguer entre deux approches à cette étude, selon que l'on se place en situation d'acteur lors de la phase de conception, ou en situation d'analyste du projet déja terminé.

D'où :

- Un mode 'opératoire' ou ensemble du savoir-faire, des règles particulières, voire des automatismes, que l'architecte utilise pour assurer une combinaison des éléments lors de la conception. La syntaxe n'est alors que le résultat de la manipulation d'un certain nombre d'opérateurs. Elle est agencement des éléments dans l'espace, mais visible uniquement 'après-coup'. Elle est contenue à l'état latent dans les opérateurs de conception mais ne guide pas leur enchainement.
Elle est 'produit' et non opérateur (Le mode opératoire ou des méthodes est l'objet du chapitre suivant).

- Un mode analytique, pouvant être parfois une véritable grammaire de ré-écriture, permettant de décomposer un projet donné au moyen d'un arbre syntaxique, pour remonter syntagme après syntagme, jusqu'à la sémie initiale. C'est cette syntaxe qui est utilisée en reconnaissance de scènes architecturales.

De quoi se compose ce type de grammaire ?

Tout d'abord il est nécessaire de bien saisir que la syntaxe architecturale n'est pas généralisée. De même qu'il existe plusieurs langues avec des syntaxes spécifiques, il existe plusieurs langages architecturaux et donc plusieurs syntaxes selon les courants, les individus, les écoles, les théories.

Le mode analytique est organisé autour de quatre pôles principaux:

- Inventaire du vocabulaire terminal (ou sémie de base) et classification catégorielle.
- Définition d'un méta-langage apte à la formalisation des règles.
- Ecriture des règles de dérivation au moyen du méta-langage et du vocabulaire terminal par analyse de projets ou de bâtiments issus d'une même théorie architecturale.
- Définition des lois de variations sur les syntagmes ou la sémie dans des contextes précis formalisés par certaines règles.

On dispose alors d'un outil d'analyse de scènes architecturales capable de décidabilité, dans une certaine mesure, sur le caractère 'bien formé' (syntaxiquement) d'une composition. Ces outils sont utilisées pour repérer dans un projet les sèmes selon le contexte de leur utilisation. Mais ne permettent pas à notre sens, d'être véritablement opérationnels dans la phase de conception, si ce n'est pour renseigner l'architecte ponctuellement, en raison de l'absence ou de la quasi absence de sémantique, et d'opérateurs qui les caractérise.

Les éléments d'Architecture possèdent un environnement sémantique complexe qu'il sera necessaire de préserver dans les futurs logiciels de C.A.O. Cet environnement peut être partiellement exprimé par la notion de champs dans les langages objets.

6. Le mode opératoire ou la notion de méthode.

Les règles pseudo-syntaxiques utilisées par les architectes lors de la conception, sont basées sur des éléments pivots (macro-éléments ou éléments de base) porteurs d'un type d'interprétation syntaxique (ensemble d'opérateurs) pour résoudre un problème ponctuel et parvenir au but fixé. (un élément pivot bien connu, est la colonne initiale qui détermine en grande part la conception du temple en architecture classique).

Nous nommerons ces opérateurs, des méthodes, terme recouvrant davantage un 'savoir-faire', une 'stratégie' du 'comment faire', qu'un procédé d'analyse ou de décidabilité d'un hypothétique caractère 'bien-formé'.

Gardons à l'esprit que les méthodes sont intimement liées et dépendantes des éléments qu'elles manipulent (éléments composant le vocabulaire que le concepteur a crée, défini et choisi pour son travail). Beaucoup sont donc différentes d'un concepteur à l'autre, et bien que l'on trouve partout un ensemble de méthodes faisant appel à des règles universellement indéfectibles, car découlant de la résistance des matériaux, ces méthodes sont souvent différentes car si les règles sont alors identiques chez tous les architectes, ce qui diffère, c'est la manière de les utiliser et le choix dans les réponses possibles.

Bien que leur structure soit complexe, elles font très certainement appel à un ensemble fini de primitives.

La recherche de cet ensemble de primitives, tâche difficile, représente un enjeu considérable en C.A.O car cette découverte permettra dans un avenir peut-être proche, de disposer d'un véritable langage de définition et de manipulation en conception.

Il est possible dès à présent de distinguer 4 grands groupes de primitives:

- Les primitives de manipulation ; telles translater, copier, afficher, détruire, effacer, mettre au nu de, mettre au droit de, mettre dans le fruit de, etc...

- Les primitives de modification du prisme signifiant ; nous entendons par là, un changement dans l'expression de l'élément par ses signifiants sans que, ni la sémantique de l'élément, ni ses

signifiants ne soient modifiés. Mais que la communication entre l'homme et l'ensemble des signifiés s'établisse sur une plémie relative différente. Changement dans la manière dont l'élément s'exprime. Tel est le cas du changement d'échelle mais non d'une opération de grossisement global des symboles du projet (zoom). Les éléments peuvent être également référencés par un numéro, un prix, un ensemble de cotes etc...

- Les primitives de modification du signifié ; le concepteur peut agir sur un aspect sémantique d'un élément au moyen d'une modification sur ses signifiants. Cette caractéristique, délicate à mettre en oeuvre et à contrôler (en effet, jusqu'où peut-on désarticuler les signifiants d'un élément en lui reconnaissant la conservation de l'appartenance à une classe sémantique donnée ?), est pourtant fondamentale en Architecture. Un opération telle que 'mettre à la proportion de' fait partie de ce groupe.

- Les primitives de localisation et sélection ; elles permettent de repérer spatialement un ensemble d'éléments à partir d'un critère donné ; 'trouver les éléments au nu de', 'trouver les éléments dans l'axe de', 'les voisins de', 'qui est au dessus de', 'l'architecte vient-il de selectionner un élément et si oui, lequel' etc... On retrouve ici l'aspect de 'syntaxe de localisation', pour laquelle un grand nombre de théoriciens de l'Architecture et de programmeurs d'analyseurs de scènes ont déja travaillé.
  C'est un mécanisme d'instanciation qui ne devrait pas poser trop de problèmes.

Cette liste n'est pas close, mais elle fournit un axe de recherche qui reste à développer.
Parallèlement à ces primitives fournissant un vocabulaire terminal pour un langage de définition de méthodes, une syntaxe devra être élaborée. Il nous semble que l'expression générale d'une méthode doit être articulée autour de deux concepts fondamentaux : Tout d'abord chaque méthode est spécifique à une classe d'éléments, car toute méthode fait appel à une connaissance de l'élément manipulé modulant sa propre fonctionnalité (la translation d'un poteau sur une trame utilise un vecteur différent de celui d'un panneau de remplissage, par exemple). D'autre part, toute méthode ne s'applique que lorsqu'un certain contexte est réalisé. Contexte qui peut être par ailleurs, très variable. (Le concepteur peut réaliser ce contexte en demandant explicitement l'application d'une méthode, mais ce peut être aussi la combinaison de certains éléments à un instant donné, etc...).

C'est pourquoi une méthode pourrait avoir une forme du type :

QUAND P ALORS A

où P est une clause prédicative, et A une procédure.

L'ensemble des méthodes dont les classes possèdent des instances (des objets) dans le projet en cours de conception, devra être en permanence 'scanné' par le programme, pour déterminer quand une méthode, de l'état passif, passe à l'état actif (instanciation de la partie gauche avec conservation des objets instanciés par unification sur des paramètres formels).
Ne sous-estimons pas la complexité d'un tel système, sachant d'une part que, les problèmes de conflits entre méthodes sont nombreux et difficiles à résoudre par programme, et que d'autre part il est ardu de solutionner le problème de la 'trace des actions' (mémorisation) quand on désire défaire ce qu'une méthode a produit .

En effet, certaines méthodes ont la capacité de venir rajouter et donc de créer, des instances de classes, permettant une sorte articulation architecturale entre éléments. C'est le cas par exemple dans un système constructif de l'élément 'joint' quand deux panneaux se trouvent côte à côte. La classe 'jointA' cherche à repérer au moyen de la partie gauche d'une méthode deux panneaux côte à côte de la classe 'panneauA'. Dès qu'une instanciation est possible, La méthode crée des instances 'jointA' et les positionne. Jusqu'à présent pas de problème, mais qu'advient-il si le concepteur décide de détruire ou de translater l'un des 'panneauA' ?

Nous avons trois possibilités: soit c'est au concepteur de retirer le 'jointA', soit la classe 'PanneauA' possède une méthode lui indiquant qu'un 'panneauA' seul ne doit pas posséder un 'jointA',

et doit donc le détruire dans le cas contraire, ou c'est la classe 'JointA' qui fait ce travail.
Car il est illusoire de penser que d'une méthode M, on puisse déduire la méthode anti-M par une procédure. Si ce travail s'avère être possible parfois, les cas sont rares, les répercussions de l'application d'une méthode sur l'environnement pouvant être considérables (Peut-être une exploration de la notion de 'stream' appliqué aux objets, fournirait-elle des pistes de solutions...).

7. Axes d'implantation.

a) Articulation du logiciel.

1- Définition de l'environnement et des classes dans un langage objet.	<=>	Logiciel de D.A.O. Constitution des symboles à partir de plèmes absolus.
2- Exploitation du système défini. Constitution d'un projet.	<=>	Logiciel de manipulation de symboles graphiques.
3- Exploitation du projet terminé. Echelles, fonctions d'agrégat sur les facettes, descriptif, quantitatif, coûts etc...		

b) Axiomes de base.

- La méta-classe est classe de toutes les classes.
- Toute classe hérite des méthodes et des facettes de ses sur-classes.
- Une classe est définie par un nom, un ensemble de facettes (champs constants, variables, procéduraux), deux champs spéciaux définissant un attachement à sa surclasse et à ses sous-classes ou ses objets, et un ensemble de méthodes.
- Un objet est défini par un ensemble de facettes et un lien sur sa classe.
- Tout objet est capable d'effectuer les méthodes de sa classe et de ses sur-classes par héritage, et seulement celles-là. Il hérite en outre des facettes de sa classe.

c) Structure proposée.

Classe	Nom	: Chaine de caractères
	Type	: Classe
	Facettes	: Ensemble de facettes
	Méthodes	: Ensemble de Méthodes
	Fils-de	: Pointeur sur la Sur-classe
	Père-de	: liste des sous-classes ou objets
Objet	Type	: Objet
	Facettes	: Ensemble de facettes
	Fils-de	: Pointeur sur sa classe

Facette	Nom	: Chaine de caractères
	Type	: Constante I Variable I Procédure
	Filtre	: Prédicat (cas des variables lors des transmissions de messages)
	Expression	: Corps de procédure ou valeur
Méthode	Déclancheur	: Clause
	Fonctionnalité	: Procédure

8. Conclusion.

Partis du plème absolu pour aboutir à la notion de méthode, nous pensons avoir présenté avec suffisament de clarté certains points de ce que nous entendons par langage architectural dans une optique systémique et structuraliste. Le présent article ne prétend pas ouvrir (re-ouvrir) un débat, souvent houleux, sur le sens et la problématique de l'Architecture face à la dure discipline du 'projet', mais plutôt d'aider les concepteurs de programmes de C.A.O à s'orienter vers de nouvelles voies, de nouvelles pistes. Nous pensons en effet, que tant que les programmes ne permettront pas aux concepteurs de définir avec précision d'une part, les éléments d'Architecture, le vocabulaire avec lequel ils désirent projetter (notions de champs, de facettes sémantiques - les références, les prix, les poids, les fournisseurs, les cotes,les matériaux etc...-, de signifiants iconiques - prise en compte de symboles spécifiques pour un même élément à différentes échelles -, de plèmes relatifs - cas des macro éléments descriptifs, tissés de relations de localisation avec les éléments qu'ils refèrent-, macro-éléments conceptuels, lois de variation, etc...), et d'autre part les méthodes qu'ils attribuent à ces éléments, les architectes risquent de trouver dans la C.A.O un apauvrissement de leur Art, du à une incapacité de communication.

En un siècle d'explosion du message, gardons lui toute sa richesse et son sens, et que les difficultés du moment ne nous fassent pas tomber dans les rêts d'une expression minimaliste mais nous montre au contraire, ce qu'il faut dépasser.

Auteur : Patrick Pérez

Laboratoire d'Informatique Appliquée à l'Architecture
Ecole d'Architecture de Toulouse
Chemin Aristide Maillol
31100 Toulouse-Mirail

Bibliographie.

R.Jakobson
"A la recherche de l'essence du langage". Gallimard.

J.Pohl
"Le symbole, clef de l'humain". S.O.D.I.

N.Chomsky
"Syntactic structures". The Hague.

"Approche transformationnelle de la syntaxe". Actes du 3ème colloque sur l'analyse linguistique. Texas 62.

C.Alexander
"Notes on the synthesis of form". Harward Univ. Press.

"Systems generating systems". Arch.Design. Oct 68.

C.Alexander, C.Ishikawa, Silverstein
"A pattern language which generates multi-service centers". Berkeley press.

C.Alexander, Poyner
"The atoms of environmental structures". Cambridge. MIT Press.

A.Churruga
"Un langage Minimal". Intern. Language Review. March 66.

H.Lefèbvre
"Le langage". P.U.F

E.Buyssens
"Les langages et le discours". O.P.

J.P.Epron
"L'architecte et la règle". Mardaga.

J.Summerson
"Le langage classique de l'architecture". L'equerre.

F.Fichet
"La théorie architecturale à l'age classique". Anthologie critique. Mardaga.

C.Norberg-Schulz
"Système logique de l'architecture". Mardaga.

K.H.Götz
"Ossatures en bois". P.P.R.-Moniteur (construire en bois).

D.Hoor
"Construction avec panneaux". P.P.R.-Moniteur (construire en bois).

J.Wheatley
"Language and rules". Berkeley Press.

G.Ryle
"Ordinary language". Cliffs.64.

E.Charniac

"Artificial Intelligence Programming". Lawrence Erlbraum Associates Publishers, 1980.

P.Cointe

"Une extension de VLISP par les objets". Science of computer Programming, 161 North Holland 1984

Y.Cullen

"Expert Systems in Architectural and Planning Education" Proceeding of the eCAADe, BRUSSEL 1983.

J.N.L.Durand

"Précis des leçons d'architecture données à l'Ecole Royale Polytechnique". Ed. Verlag Dr. Alfons Uhl, Nördlingen, 1981.

M.Duplay

"Methode illustrée de conception architecturale." Editions du Moniteur. 1983.

A.Goldberg

"SMALLTALK-80". Addison Welsley Pub, 1983.

J.P.Goulette

"Le dessin d'Architecte et l'Informatique". Travail personnel de fin d'études. Ecole d'Architecture de Toulouse, Novembre 1984

LI2A-83

"Les utilitaires et l'intelligence artificielle pour un système d'aide à la conception en Architecture" Rapport final d'une recherche financée par le Secrétariat de la Recherche Architecturale. LI2A Juin 1983.

LI2A-84

"Les utilitaires et l'intelligence artificielle pour un système d'aide à la conception en Architecture" Rapport final d'une recherche financée par le Secrétariat de la Recherche Architecturale. LI2A Avril 1985.

LI2A-85

"Les utilitaires et l'intelligence artificielle pour un système d'aide à la conception en Architecture" Rapport final d'une recherche financée par le Secrétariat de la Recherche Architecturale. LI2A Décembre 1985.

LI2A-84
D.Caradant

"Les utilitaires et l'intelligence artificielle pour un système d'aide à la conception en Architecture". Rapport final de recherche. LI2A, Juin 1984.

LI2A-85
J.P. Goulette
P. Pérez

"Tangram, Manuel de l'utilisateur", in "Aides Intelligentes au dessin d'Architecte", Rapport final de la Convention Adl 84/935, LI2A, Sept. 85.

M.Minsky

"A Framework for Representing Knowledge". in The Psychology of Computer Vision, McGraw-Hill, NY, 1975.

Waterman
Hayes-Roth

"Pattern-Directed Inferences Systems". Ed Academic Press, 1977.