



**HAL**  
open science

## The SEPT application: Appraisal of an experiment

Patrick Brezillon

► **To cite this version:**

Patrick Brezillon. The SEPT application: Appraisal of an experiment. [Research Report] lip6.2002.008, LIP6. 2002. hal-02545608

**HAL Id: hal-02545608**

**<https://hal.science/hal-02545608v1>**

Submitted on 17 Apr 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# L'APPLICATION SEPT: BILAN D'UNE EXPÉRIENCE

**Patrick Brézillon**

*LIP6, case 169, Université Paris 6*

*8 rue du Capitaine Scott,*

*75015 Paris*

*France*

[Patrick.Brezillon@lip6.fr](mailto:Patrick.Brezillon@lip6.fr)

---

**ABSTRACT:** The paper presents the result of a seven-years experiment with the SEPT application (SEPT is the French acronym for equipment monitoring in an extra-high voltage substation). In this framework several software pieces have been developed for different reasons: a diagnoser, a simulator and a knowledge configurator. Moreover, explanation generation has been considered along two different ways: built from a local model of the reasoning, and built under the responsibility of the operator. We discuss the different steps of the work and the rationale behind the choices made at that time. This application permits to point out the essential role of the context, and the natural movement towards the context-based intelligence assistant systems.

**RÉSUMÉ** Ce papier dresse le bilan d'une expérience de sept ans avec l'application SEPT (Surveillance d'Équipements dans un Poste à très haute Tension--400.000 volts) dans laquelle plusieurs systèmes ont été réalisés pour diverses raisons: un diagnostiqueur, un simulateur et un configurateur de connaissances. En outre, la génération d'explication a été considérée de deux manières différentes et originales: d'une part à partir d'un modèle local du raisonnement, et, d'autre part, en laissant l'opérateur guider la génération des explications. Nous discutons les différentes étapes de ce travail et les raisons des choix effectués. Cette application permet de mettre en évidence le rôle essentiel que joue le contexte, et la migration naturelle du type de système qui fut développé vers les systèmes d'assistance intelligents basé sur le contexte.

**KEY WORDS:** *Expert systems, explanation, knowledge configuration, simulation, context, intelligent assistant systems.*

**MOTS-CLÉS:** *Systèmes experts, explication, configurateur de connaissances, simulation, contexte, systèmes d'assistance intelligents.*

---

## 1. Introduction

Le système expert SEPT (Surveillance d'Équipements dans un Poste à très haute Tension--400.000 volts) a été conçu et développé en collaboration avec EDF sur une période s'étendant de 1987 à 1993. Ce diagnostiqueur réalisé, les travaux ont suivi deux voies en parallèle. La première a consisté à réaliser un simulateur et à le coupler avec le diagnostiqueur. L'objectif était de valider SEPT de la manière la plus exhaustive possible car les incidents sur le réseau électrique français sont rares et les corrections apportées très rapidement. Les recherches sur la voie parallèle ont porté sur le développement de véritables capacités explicatives dans SEPT en étendant les fonctionnalités du formalisme de représentation initialement choisi. L'idée était de fournir à l'utilisateur les moyens de guider la génération d'une explication par le système. Finalement, il s'est avéré difficile d'installer SEPT dans un nouveau poste à THT sans réécrire SEPT de manière assez profonde (principalement dans les règles d'expertise). Ceci nous a conduit à développer le système KA7, un système de configuration automatique de systèmes experts. Dans l'ensemble, cette application a donné lieu à deux thèses et cinq stages de DEA. Toutes les publications faites dans le cadre de cette application sont données à la fin de cette présentation.

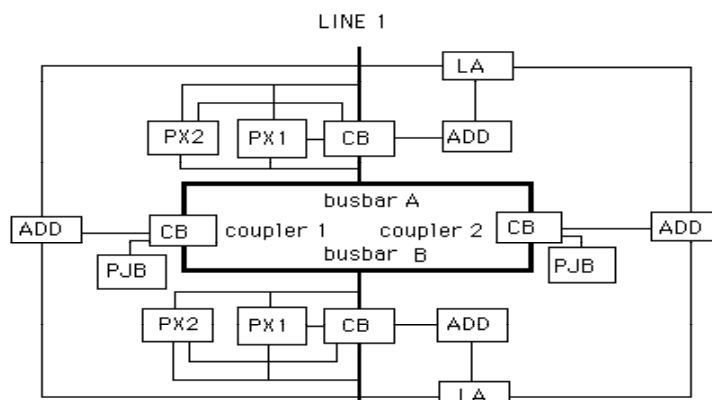
Nous présentons ci-après les différentes étapes entrant dans le cadre de l'application SEPT. La Section 2 fait tout d'abord un bref rappel sur les connaissances du domaine, principalement en ce qui concerne la topologie d'un poste à très haute tension, le diagnostic des systèmes de contrôle. La Section 3 décrit les connaissances du diagnostiqueur, leur représentation, et la modélisation qui est faite du contrôle. La Section 4 discute du couplage du diagnostiqueur et du simulateur, en commençant par le rôle du simulateur, et l'architecture finale obtenue. La Section 5 introduit la manière dont la génération d'explications avait été envisagé dans le diagnostiqueur, et les modifications apportées au langage de représentation des connaissances et du raisonnement. La section 6 introduit le configurateur de connaissances qui a été conçu et développé afin d'assurer la portabilité du diagnostiqueur. Finalement, la section dresse le bilan de cette application SEPT, en discutant de l'organisation des connaissances, l'exploitation des spécificités de la machine pour raisonner, l'importance de la simulation, le rôle pivot que joue le contexte, et l'architecture qui se dégage pour un véritable système d'assistance intelligent en contexte. Toute la bibliographie public relative à l'application SEPT est regroupée dans la section 8 avec les références citées dans le corps du papier.

## 2. Connaissances du domaine

### 2.1. Topologie des postes à THT (Très Haute Tension)

Le réseau électrique très haute tension (400 000 volts) est un réseau fortement maillé qui est constitué de lignes (en triphasé) reliées entre-elles par des nœuds, les postes à très haute tension. La Figure 1 montre le schéma simplifié d'un poste à THT qui est composé de jeux de barres (busbars) et de transformateurs qui sont interconnectés par l'intermédiaire de systèmes de contrôle (SC). Chaque SC regroupe différents équipements comme les disjoncteurs (CB), des protections de distance (PX1 et PX2), et d'autres équipements (ADD, LA, etc.). Les lignes sont connectées aux jeux de barre par l'intermédiaire de ces systèmes de contrôle. Les jeux de barres sont connectés entre eux par l'intermédiaire de systèmes de contrôle particuliers (couplers) constitués d'équipements tels qu'un disjoncteur (CB) et une protection de distance (PJB).

Les éléments d'un réseau électrique sont sujets à des défauts appelés courts-circuits. Un court-circuit peut être transitoire ou permanent, résistant ou non-résistant, monophasé ou polyphasé. Le rôle d'un système de contrôle est de diagnostiquer et d'éliminer les courts-circuits sur le réseau électrique par l'ouverture automatique de certains disjoncteurs déclenchés par des protections de distance.



**Figure 1:** Vue simplifiée d'un poste à très haute tension

Il y a une haute redondance voulue dans le fonctionnement des équipements afin de pallier les éventuelles défaillances des équipements. Ainsi, une faute peut être éliminée à différents niveaux, d'une part, si une protection de distance ne détecte pas un défaut, l'autre protection le fera, et, d'autre part, si un disjoncteur refuse de s'ouvrir, alors les disjoncteurs avoisinants interviendront. Ainsi, à l'apparition d'une faute, c'est tout un ensemble de systèmes de contrôle qui "voient" la faute et réagissent, mais normalement seuls les systèmes de contrôle qui encadrent directement l'élément du réseau supportant la faute doivent s'ouvrir. Techniquement, le fonctionnement des systèmes de contrôle est réglé selon la définition d'une zone de protection qui permet le réglage de temporisations de fonctionnement des équipements dans les systèmes de contrôle. Par exemple, il est prévu une telle temporisation entre l'ordre de déclenchement émis par une protection de distance et l'ouverture du disjoncteur pour que le disjoncteur ne s'ouvre pas si la faute disparaît toute seule entre temps. En conséquence, cette redondance assure que toute faute apparaissant sur le réseau sera éliminée d'une manière ou d'une autre.

Une information contextuelle importante est que les équipements dans les systèmes de contrôle ne fonctionnent tant qu'ils voient la faute. Ainsi, le diagnostic d'une défaillance d'équipement (ou d'une incohérence entre le fonctionnement de deux équipements) s'effectue par analyse du fonctionnement des équipements pendant l'élimination d'une faute. Le fonctionnement d'un équipement est caractérisé par une suite de changements d'états (par exemple, envoi d'un ordre de déclenchement au disjoncteur). Un changement d'état est appelé événement, et chaque événement (par exemple, l'émission de l'ordre de déclenchement) fait l'objet d'un signal qui est enregistré sur un consigneur d'état qui centralise ainsi l'ensemble des événements qui surviennent suite à l'apparition d'une faute. Beaucoup d'équipements voyant la faute, il y a donc une avalanche de signaux qui arrivent au consigneur d'état pendant un laps de temps très courts (généralement 1-2 secondes), avec des problèmes comme le télescopage de signalisations (simultanéité de deux signaux qui entraîne qu'un seul est enregistré sur le consigneur d'état).

C'est sur la base des informations recueillies sur le consigneur d'état que le diagnostic est fait. Le diagnostic est mené en comparant la séquence logique des événements et la séquence réelle des événements qui se sont produits : au niveau de chaque équipement, au niveau des équipements dans un système de contrôle, et au niveau du poste à THT. Ceci permet de vérifier différentes cohérences de fonctionnement. Par exemple une protection de distance qui voit un défaut à  $T_0$  envoie au disjoncteur un ordre de déclenchement à  $T_0 + \text{Temporisation}$ . L'existence d'un ordre de déclenchement suppose que la protection ait vu un défaut (enregistrement d'un signal préalable de la protection sur le consigneur

d'état), que l'ordre de déclenchement soit suivi à  $T_0 + \text{Temporisation}$  d'un signal d'ouverture du disjoncteur (cohérence au niveau du système de contrôle).

## 2.2. Diagnostic des systèmes de contrôle

Les signaux arrivant sur le consignateur d'état donnent une image relativement fidèle du fonctionnement des équipements dans le poste à THT. Néanmoins, ces signaux ne permettent pas une identification directe de la faute sur le réseau qui est à l'origine de la réaction des systèmes de contrôle. Les caractéristiques de la faute doivent être inférées à partir des événements qui ont été enregistrés sur le consignateur d'état. Le diagnostic doit donc être de type *bottom-up* en quatre étapes successives:

- Vérification du fonctionnement individuel des équipements dans le poste,
- Comparaison du fonctionnement des équipements au niveau de chaque système de contrôle,
- Etablissement de la cohérence entre les fonctionnements des systèmes de contrôle au niveau du poste
- Confrontation des réactions des équipements du poste avec le comportement des systèmes de contrôle des postes distants.

## 3. Connaissances du diagnostiqueur SEPT

### 3.1. Modélisation des connaissances de diagnostic

La réalisation du système expert SEPT a initialement été faite dans un langage écrit à EDF pour développer des systèmes experts (Mulet-Marquis et Gondran, 1985). Les fonctionnalités de ce langage ont ensuite été étendues afin de pouvoir manipuler explicitement des paquets de règles afin de rendre compte de la structuration des connaissances du domaine, et afin de pouvoir "rejouer" un diagnostic avec différents niveaux de détails et différents points de vue pour générer des explications sur une notion d'interprétation.

SEPT mène un diagnostic en deux temps: il réalise un diagnostic initial, puis il fournit une explication des événements basée sur un modèle. Le modèle est utilisé dans SEPT sous une forme de connaissances compilées, d'une part, pour le diagnostic et, d'autre part, pour les explications. Il décrit les interactions en termes de séquences de signaux événementiels avec les diverses conditions qui indiquent les états du système comme les caractéristiques de la faute, les informations structurelles, etc. Un exemple d'une telle séquence d'événements est donné dans la Figure 2.

SI	une faute est non-résistante et que l'équipement ADD d'un coupleur émet le signal défaillance disjoncteur DF.BARRE au temps T,
ALORS	Le disjoncteur doit s'ouvrir en triphasé et émettre l'ordre DJ.OUV correspondant à 40 ms après l'instant T, en supposant que le disjoncteur soit encore fermé, n'a pas de défaillance et qu'il n'y a pas eu de signal manque de polarité.

*Figure 2: Exemple de règle décrivant une séquences d'événements logique*

Il est à noter que la plupart des règles dans SEPT contiennent des restrictions d'utilisation de la règle (par exemple, "en supposant que le disjoncteur soit encore fermé") qui définissent en fait le contexte d'utilisation de chaque règle. Pour le diagnostic initial, il est adopté un raisonnement de type associatif où les connaissances sur le diagnostic sont compilées à partir du modèle de contrôle. Ce diagnostic aboutit à une première sortie sur les mauvais fonctionnements intrinsèques des équipements, les caractéristiques de la faute, et les incohérences dans le fonctionnement des équipements. Une contrainte à respecter était de fournir le plus rapidement possible un résultat synthétique.

### 3.2. Modélisation du contrôle

Pour l'explication basée sur le modèle, les deux principales tâches sont la prédiction et une post-analyse. L'étape de prédiction vise à donner une vue des événements dans le futur immédiat. L'étape de post-analyse permet d'inférer l'évolution du système à partir de séquences d'événements et d'expliquer cette évolution en la comparant aux observations du modèle de contrôle.

Une séquence d'événements avec ses diverses conditions est généralement exprimée comme l'apparition du signal de sortie au temps " $T_0 + \text{Temporisation}$ " qui est causé par l'apparition du signal d'entrée au temps  $T_0$  et de la validité de diverses conditions à ce temps  $T_0$  (préconditions) et au temps  $T_0 + \text{Temporisation}$  (postconditions). La Figure 3 illustre la représentation d'une telle séquence d'événements.

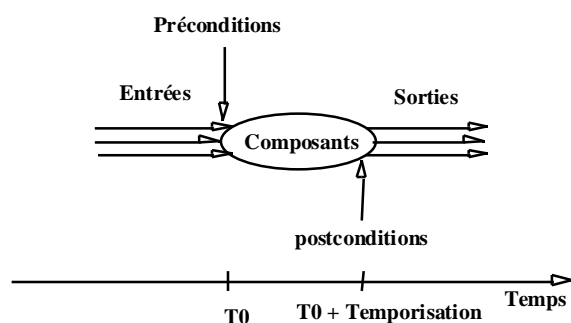


Figure 3: Eléments intervenant dans une séquences d'événements

C'est sur la base de ces éléments que l'on peut s'assurer de la validité des informations sur le consignateur d'état (par exemple, détection d'informations manquantes ou incohérentes), et que l'on peut expliquer le décours réel des événements par rapport à ce qui manque ou qui est faux. Il est important de noter ici que le modèle sur lequel sera basé les explications du diagnostic est ni un modèle de bon fonctionnement, ni un modèle de mauvais fonctionnement. Il s'agit du modèle du fonctionnement tel qu'il s'est effectivement déroulé localement (bon fonctionnement et dysfonctionnement mélangé). En effet, il s'agit d'un modèle local (prédiction sur peu de pas en avant et post-analyse limitée). Les explications sont donc locales, pas comme un diagnostic global vu comme la construction d'un modèle spécifique de la situation dans son ensemble (Clancey, 1992).

### 3.3. Prédiction et post-analyse

La prédiction d'une séquence d'événements doit prendre en compte la perte éventuelle de signaux au niveau du consignateur d'état. Pour déterminer de combien en avant doit être faite la prédiction, il y a un compromis pratique à trouver entre le temps d'exécution et les besoins en explication sur un signal perdu.

6

La post-analyse consiste à:

- 1) identifier la séquence d'événements qui a été prédite,
- 2) identifier tout événement inopportun (n'appartenant pas à la prédiction), et
- 3) expliquer les comportements anormaux identifiés lors du diagnostic initial.

Le module d'explication basée sur le modèle:

- 1) commence une session d'explication par la présentation de la topologie initiale du réseau et de la faute qui a été identifiée dans le diagnostic initial;
- 2) prédit et présente les équipements normalement activés ainsi que la zone de protection couverte par les systèmes de contrôle,
- 3) classe chronologiquement tous les signaux,
- 4) à chaque changement d'état d'un équipement, fait une post-analyse et explique l'état courant du système à partir de ses prédictions précédentes et des événements réels survenus, prédit quel doit être le comportement des systèmes de contrôle dans le futur proche, et explique tout comportement anormal qui a été identifié dans les systèmes de contrôle à ce moment,
- 5) retourne à (3) pour expliquer tous les signaux, et
- 6) rend compte de la manière dont la faute a été isolée structurellement.

La Table 1 donne un exemple d'explication produite par SEPT.

Un court-circuit non-résistant est apparu dans la première partie de LIGNE-1 au temps  $T_0 + 10$  millisecondes.

Les protections de distance PX1 et PX2 de LIGNE-1 doivent détecter la faute en zone 1;  
Les protections de distance PX1 et PX2 de LIGNE-2 doivent détecter la faute en zone 4;  
Les protections de COUPLAGE-1 et de COUPLAGE-2 doivent détecter la faute et s'orienter vers le jeu de barres A à travers lequel la faute est vue.

T =  $T_0 + 30$  ms (signalisation d'événements)

1. LIGNE-1	PX1.ZONE1	$T_0 + 30$ ms
2. LIGNE-2	PX1.ZONE1	$T_0 + 30$ ms
3. LIGNE-2	PX2.ZONE1	$T_0 + 30$ ms
4. COUPLAGE-1	PJB.DIR.A	$T_0 + 30$ ms

Séquence correcte de signalisation d'événements:

La protection de distance PX1 de LIGNE-1 détecte la faute en zone 1.

La protection PJB de COUPLAGE-1 détecte la faute et s'oriente vers le jeu de barres A.

Séquence inopportune de signalisation d'événements:

Il y a activation intempestive par mauvaise identification de zone des protections de distances PX1 et PX2 de LIGNE-2 qui détectent la faute en zone 1 au lieu de zone 4.

T =  $T_0 + 71$  ms (signalisation d'événements)

5. LIGNE-2	DJ.OUV	$T_0 + 71$ ms
------------	--------	---------------

Séquence correcte de signalisation d'événements:

SI une protection de distance PXi de LIGNE-2 émet un signal PXi.ZONE1 au disjoncteur à T, ALORS le disjoncteur devra s'ouvrir en triphasé en émettant le signal DJ.OUV 40 ms après

T, en supposant que le disjoncteur soit encore fermé à ce temps et qu'il ne présente pas de défaillance.

Comportement anormal

Des comportements parallèles ne sont pas respectés: le signal PX1.ZONE1 de PX1 (LIGNE-1) doit être accompagné du signal PX2.ZONE1 de PX2.  
PJB (COUPLAGE-2) n'a pas émis le signal PJB.DIR.A et ne s'est pas orientée vers le jeu de barres A où la ligne en faute LIGNE-1 est connectée.

etc.

*Table 1: Exemple d'explication dans SEPT*

### 3.4. Discussion

Les deux étapes du diagnostic sont complémentaires: le diagnostic initial donne une synthèse des événements, et l'explicateur basé sur un modèle fournit une explication causale profonde de la séquence des événements qui se sont déroulés depuis l'apparition de la faute jusqu'à son élimination par les diverses actions de contrôle des équipements.

Dans cette approche, il n'est pas nécessaire de disposer d'une bibliothèque exhaustive d'anomalies (ce qui n'est d'ailleurs pas possible vu le nombre et la variété d'anomalies possible), ni même de conservées toutes celles qui sont traitées par le système. De plus, le cas de fautes multiples sur le réseau est naturellement traité dans l'approche décrite ci-dessus.

SEPT a été écrit dans un formalisme de règles de production basé sur la logique des prédicats du premier ordre à l'aide du générateur de systèmes expert développé à EDF. Sans altérations de la base de connaissances, le diagnostiqueur a permis de résoudre avec succès tout un ensemble d'incidents survenus dans des postes à THT de topologies différentes. SEPT a été installé pendant une année pour validation dans un poste à THT de la région parisienne (il a même trouvé une défaillance d'un équipement qui n'avait pas été repérée par les opérateurs humains).

Toutefois, SEPT présentait des faiblesses, la principale étant une représentation artificielle de la structuration des connaissances (principalement les connaissances sur la topologie du poste) à l'aide de clauses écrans, structuration qui revenait à faire du procédural dans un langage déclaratif. Deux actions ont alors été entreprises. Une première tentative a été de développer SEPT dans un environnement permettant l'association de plusieurs formalismes de représentation des connaissances et du raisonnement (Knowledge Craft). La deuxième tentative a consisté à étendre le formalisme à base de règle de production initial afin de supporter la représentation explicite de la structuration des connaissances sous la forme de paquets de règles indépendants les uns des autres, et la possibilité de simuler le comportement du diagnostiqueur dans différents contextes pour obtenir des explications à différents niveaux de détails.



## 4. Couplage du diagnostiqueur avec un simulateur

### 4.1. La nécessité d'un simulateur

SEPT a été initialement développé pour accomplir une tâche de diagnostic, partie du système que nous avons appelé diagnostiqueur. À l'inverse de l'application SART pour le métro (Brézillon et al., 1997), il n'a pas été possible de le valider sur un ensemble très large de situations incidentielles car le nombre d'incidents sur le réseau EDF est très faible et leur variété pratiquement inconnue. Pour remédier à cet état de fait, il a été décidé de concevoir et développer un modèle fonctionnel du réseau électrique, appelé simulateur, afin de pouvoir décrire différentes topologies de réseaux électriques, leurs comportements, et surtout pouvoir définir différents types de défaut, soit de manière manuelle, soit de manière aléatoire pour une exploitation automatique du simulateur couplé au diagnostiqueur.

Une fois les deux systèmes réalisés séparément, s'est posé la question du couplage du diagnostiqueur et du simulateur au sein d'un système unique, tout en veillant à ce que chacun d'eux puisse être utilisé seul, indépendamment de l'autre. Deux raisons nous ont amenés à concevoir une architecture unique: (1) pour une validation exhaustive du système résultant du couplage du diagnostiqueur et du simulateur, et (2) pour développer un réel système explicatif basé sur la simulation pour la formation des futurs opérateurs. En termes plus modernes, il s'agissait de réaliser une plate-forme où pourraient interagir deux agents, un agent diagnostiqueur et un agent simulateur.

Le problème majeur était que le diagnostiqueur aussi bien que le simulateur contenait deux types de connaissances: des connaissances de surface et des connaissances profondes. Une partie de ces connaissances reposaient sur les connaissances du domaine, connaissances qui sont communes au diagnostiqueur et au simulateur (par exemple, la topologie d'un poste à THT), bien qu'elles soient sous des représentations différentes adaptées à chaque utilisation qui est faite de ces connaissances. Par ailleurs, l'évolution constante des technologies faisait qu'il était nécessaire de prévoir dès le début des possibilités d'évolution des bases de connaissances afin d'assurer une durée de vie réaliste à SEPT lors de l'introduction de nouvelles technologies dans les postes à THT.

L'ensemble de ces raisons nous ont conduit à rechercher une architecture robuste pour supporter le couplage du diagnostiqueur et du simulateur.

### 4.2. Le couplage du diagnostiqueur et du simulateur

La littérature montre qu'il existe plusieurs architectures possibles pour associer différents types de connaissances comme les connaissances de surface et les connaissances profondes. Il s'agit là d'un problème qui était bien connu à l'époque, mais l'association de connaissances de surface et de connaissances profondes n'avaient jamais été décrites selon un point de vue "couplage" de deux systèmes. (Rappelons que cette exigence provenait de la nécessité de pouvoir utiliser séparément le diagnostiqueur et le simulateur.)

La nature particulière de notre problème (identification de la faute à partir du diagnostic, parallélisation des éléments d'un système de contrôle et redondance des équipements) nous a fortement guidé dans notre choix d'un type de couplage qui ne repose pas sur les bases généralement décrites dans la littérature.

La Figure 4 présente les cinq types de couplages identifiés dans la littérature correspondant à nos besoins. Les connaissances profondes sont représentées par la lettre S (comme simulateur). Les connaissances de surface sont représentées par la lettre D (comme diagnostiqueur). Les deux types de connaissances (profondes et de surface) sont des systèmes à base de règles qui travaillent (et interagissent) sur une base de fait (la partie avec des points) qui sert à la fois de milieu de travail et de communication.

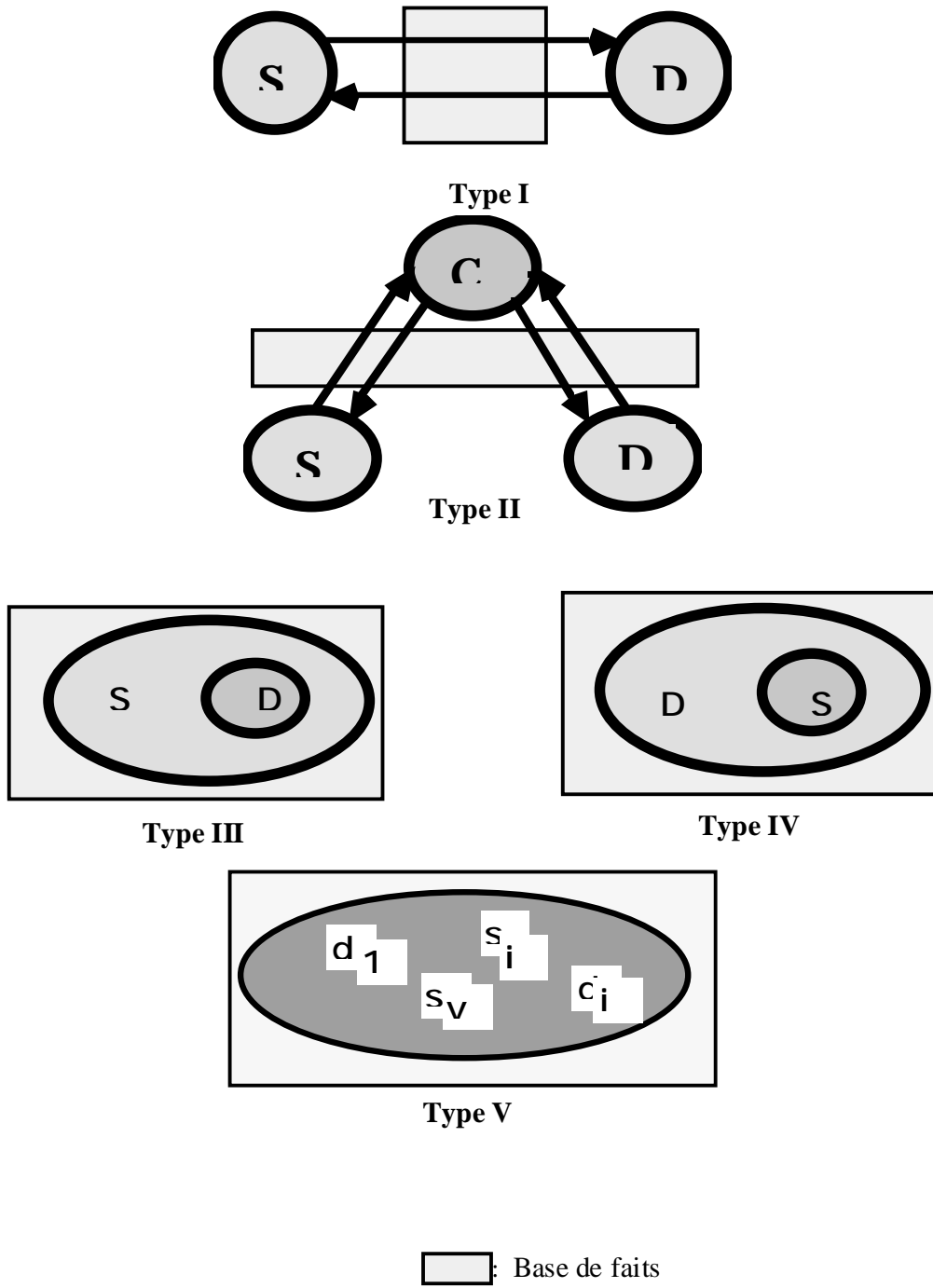
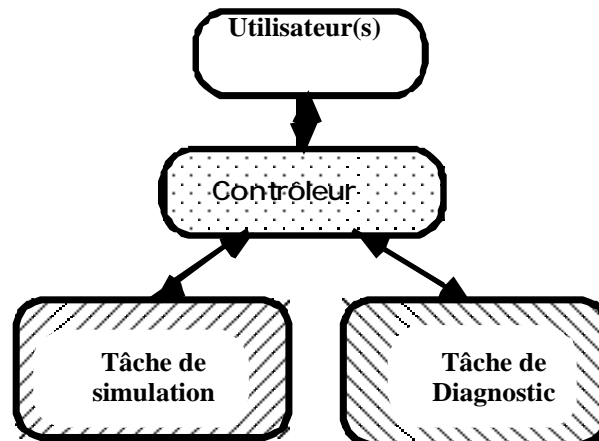


Figure 4: Les cinq types de couplage

Ces cinq types de couplage peuvent être divisés en deux grandes classes. La première classe contient les couplages de Types I et II où connaissances de surface et profondes peuvent être représentées par des systèmes autonomes, soit avec un couplage direct (Type I) ou indirect (Type II). Dans le Type II, la communication entre S et D est assurée par l'intermédiaire d'un sous-système appelé C pour contrôleur dans la Figure 4. Dans la deuxième classe de couplages, les deux corps de connaissances sont représentés par des systèmes indépendants (Types III, IV et V) où soit S soit D est une partie de l'autre, ou les deux systèmes sont atomisés au sein d'un système unique (Type V). (Ce dernier type était un cas limite pour notre propos.)

D'après la littérature, la préférence accordée à un type sur les autres dépend de la fonction qui était jugée principale dans le système (diagnostic ou simulation). Dans notre cas, les deux fonctions devant avoir le même poids, car devant pouvoir être utilisé indépendamment l'un de l'autre. Les couplages de types III, IV et V ne pouvaient donc pas satisfaire nos attentes. Notre choix se limitait donc entre les couplages de types I et II. La principale différence entre ces deux types de couplages était la présence ou non du troisième module, le contrôleur. Pour des raisons décrites par ailleurs (voir les articles autour de 1989), notre choix s'est finalement porté sur un couplage de type II comme représenté dans la figure suivante.

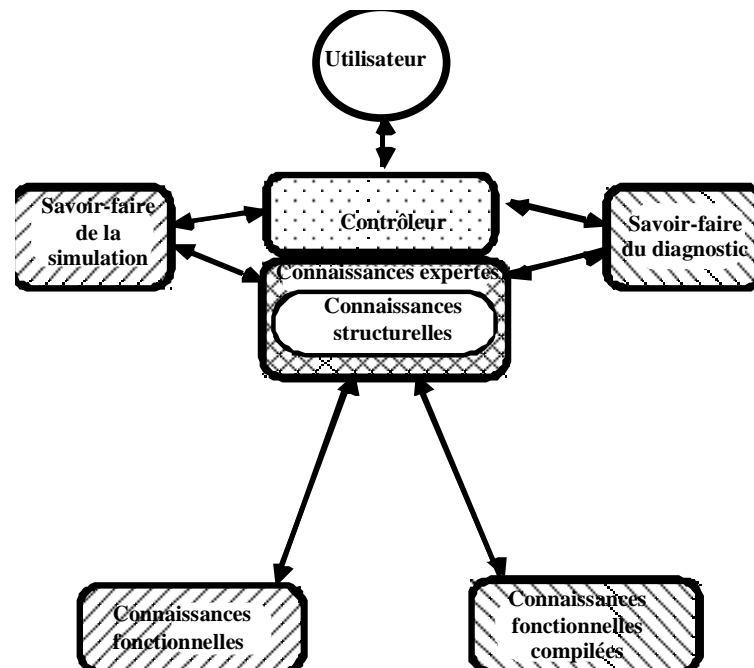


*Figure 5: Couplage du diagnostiqueur et du simulateur au niveau connaissance*

Depuis, d'autres raisons ont conforté ce choix, comme le fait que l'utilisateur ne devait avoir qu'un seul "interlocuteur", le contrôleur et évitait ainsi une surcharge cognitive liée à l'obligation de parler à chaque agent dans son langage spécifique.

#### **4.3. Une architecture uniforme pour le couplage du diagnostiqueur et du simulateur**

Le type de couplage étant choisi, nous avons eu à faire face à d'autres contraintes. Par exemple: (1) la nécessité de ne pas dupliquer les connaissances du domaine qui étaient communes au diagnostiqueur et au simulateur, (2) limiter les interventions pour la maintenance et la mise à jour des bases de connaissances, (3) la division au sein de chaque sous-système (diagnostiqueur et simulateur) de la base de connaissances en savoir-faire, compétences opérationnelles et topologie d'un poste à THT, et (4) l'intérêt d'avoir une structure ouverte facilement modifiable afin d'accepter ultérieurement d'autres modules comme un module statistique ou un module de formation. Au niveau implémentation, nous avons donc fait éclater les modules initiaux et obtenu ainsi une architecture plus homogène telle que représentée dans la Figure 5 (voir Figure 6).



*Figure 6: Couplage du diagnostiqueur et du simulateur au niveau implémentation*

SEPT, dans cette architecture, a apporté de nombreuses satisfactions. D'une part, il était possible d'accomplir la tâche de diagnostic ou de simulation seule. D'autre part, SEPT pouvait être totalement validé (automatiquement si nécessaire) par l'utilisation "en boucle" du simulateur et du diagnostiqueur où l'utilisateur entrait les caractéristiques d'une faute, choisissait une topologie pour son poste à THT, déroulait la simulation afin d'obtenir une séquence théorique des signaux enregistrés sur le consignateur d'état, puis l'utilisation du diagnostiqueur qui devait retrouver les données entrées par l'utilisateur. Enfin, l'architecture ouverte permettait dès le départ de prévoir l'introduction naturelle ultérieure de modules pour traiter de nouvelles tâches.

## 5. Génération d'explication dans SEPT

### 5.1. Introduction

La génération d'explications répondait à un besoin réel. En effet, les sorties du diagnostiqueur pouvaient être particulièrement complexes, et les liens entre les caractéristiques d'une faute et les défaillances d'équipement détectés peu évidents. Toutefois, les approches existant à l'époque dans la production d'explication ne laissaient pratiquement pas l'utilisateur intervenir dans la construction de l'explication. Or dans notre cas, l'utilisateur est un expert, pas un novice. Il ne s'agissait donc pas de lui fournir la définition d'un disjoncteur, mais les raisons qui motivaient le système de conclure à un dysfonctionnement du disjoncteur.

Au lieu de laisser le système contrôler toute la construction des explications, nous avons choisi de procéder à la démarche inverse en fournissant à l'utilisateur tous les moyens pour contrôler comment le système construisait une explication, ces moyens comprenant la possibilité de rejouer un diagnostic dans différents contextes d'explication. Pour cela nous avons développé deux fonctionnalités: la représentation explicite de paquets de règles et la notion d'interprétation.

## 5.2. Les paquets de règles

L'utilisation explicite de paquets de règles (la base de règles est en fait constituée d'un ensemble de bases de règles) a permis d'introduire une notion de contexte qui définissait les règles que le moteur d'inférence devait considérer à un instant donné. Par exemple, la Figure 7 donne un exemple de règle faisant appel à un paquet de règles de diagnostic lorsqu'un équipement est identifié comme étant un disjoncteur.

```

! check_cb
"Test du disjoncteur du système de contrôle $name"
> if failure freeze check_pw, check_teac
IF
    equipment_piece (cos) := (cb) ,
    nature (cb) := circuit_breaker .
THEN
    call the rule packet 'Circuit-Breaker_Diagnosis(cos, cb)' .

```

*Figure 7: Exemple d'appel d'un paquet de règles*

La règle *check\_cb* présentée dans la Figure 7 est écrite en pseudo langue naturelle pour faciliter la compréhension. Elle est utilisée pour déclencher le diagnostic d'un disjoncteur *cb* (un paquet de règles) dans un système de contrôle *cos*. Cette règle appartient à un paquet de règles qui teste tous les équipements dans un système de contrôle. Le paquet de règles représente l'expertise de diagnostic au niveau d'un système de contrôle, et les expertises locales de diagnostic sur les équipements sont dans d'autres paquets de règles comme *Circuit-Breaker\_Diagnosis* dans l'exemple de la Figure 7. Au déclenchement de la règle *check\_cb*, le moteur d'inférences suspend son exploitation du paquet de règles courant où se trouve la règle *check\_cb*, et prend le paquet de règles *Circuit-Breaker\_Diagnosis* avec les instances qui sont données dans la règle pour les variables *cos* et *cb*. Ainsi un paquet de règle peut être utilisé en logique des propositions lors que les instances des variables sont fixées, ou en logique d'ordre 1, ce paquet de règles pouvant être utilisé pour tous les disjoncteurs dans le poste à THT (en général une vingtaine par poste).

Dans la règle présentée dans la Figure 7, le contexte est représenté à trois niveaux: (1) par l'identification d'un équipement en tant que disjoncteur, (2) par le passage d'un jeu d'instances pour les variables qui va focaliser le diagnostic sur un disjoncteur unique, et (3) par une gestion du diagnostic au niveau du système de contrôle par la méta-connaissance *if failure, freeze choice-pw, choice\_teac* qui dit que si une défaillance est mise en évidence sur le disjoncteur, il n'est pas nécessaire de tester alors les équipements *pw* et *teac* de ce système de contrôle.

Le moteur d'inférence n'a ainsi qu'à considérer que le paquet de règles pertinent, et souvent pour un jeu d'instances données, ce qui permettait de retrouver certaines propriétés de la logique propositionnelle. Les gains d'une telle écriture sont aussi bien au niveau implémentation (rapidité des inférences) qu'au niveau compréhension des connaissances puisqu'il n'y avait pas de clauses écrans, le contrôle étant explicitement représenté sous forme de règles de production au même titre que les autres connaissances.

La notion de contexte est ici attachée aux connaissances décrites sous la forme de règles de production. Pour la génération d'explication, nous avons aussi développé un outil, appelé la gestion

d'interprétations, qui permettait à l'utilisateur de définir par lui-même le contexte dans lequel il souhaitait voir se dérouler le diagnostic.

### 5.3. La notion d'interprétation

Nous allons considérer la règle suivante qui conclut à un comportement anormal d'une protection de distance:

<b>SI</b>	l'ordre de déclenchement d'une protection de distance est émis sur deux phases,
<b>ALORS</b>	Action: Conclure à un fonctionnement anormale de la protection de distance.

*Figure 8: Exemple de règle nécessitant l'adjonction d'une explication*

SEPT doit générer des explications pour des opérateurs qui ont un haut niveau d'expertise, expertise issue de leur expérience du terrain. Leur expertise est donc plus pragmatique que théorique, à l'inverse des connaissances dont SEPT a été alimenté. Par exemple, dans la règle donnée dans la Figure 8, il est difficile de fournir à l'opérateur une explication, celui-ci sait qu'une ouverture du disjoncteur sur deux phases seulement signifie un problème sérieux car une ouverture de disjoncteur ne peut se faire que sur une phase ou les trois phases en même temps. Pour qu'une explication soit pertinente pour l'opérateur, il faut lui décrire la cause du dysfonctionnement, mais une telle connaissance n'était pas explicitée dans SEPT puisqu'elle n'est pas nécessaire pour le diagnostic.

La solution retenue a été de laisser l'opérateur construire seul son explication en lui fournissant les moyens et les connaissances nécessaires. Les moyens reposent sur une structuration explicite des connaissances du domaine pour organiser les explications et laisser l'opérateur choisir le niveau de détail souhaité pour les explications que le système doit lui fournir. La structuration des connaissances fait appel, au niveau implémentation de la base de règles, à l'organisation en paquets de règles. Au niveau des règles, l'exécution des actions en partie ALORS est contrôlée par un marqueur prédéfini I1, I2, etc. suivant le niveau de détail de l'explication associée, et le type d'explication souhaité. L'utilisateur définit alors l'interprétation sous laquelle il souhaite voir dérouler le diagnostic par le système (par exemple, I1, I3, ou I1 + I3). La même simulation peut être rejouée plusieurs fois sous différentes interprétations, donc avec des explications différentes par leur nature et leur niveau de détail. La Figure 9 reprend l'exemple de la règle donnée dans la Figure 8 avec les explications associées à la règle. (Pour ce papier, nous avons choisi le type d'explication qu'un opérateur novice pourrait demander.)

<b>SI</b>	l'ordre de déclenchement d'une protection de distance est émissur deux phases,
<b>ALORS</b>	Action: Conclure à un fonctionnement anormal de la protection de distance;
I1	Explication de niveau 1: Envoyer un message d'avertissement;
I2	Explication de niveau 2: Présentation des prémisses instanciées;
I3	Explication de niveau 3: Une ouverture sur deux phases signifie probablement que la

	faute concerne trois phases et que la protection est défailante sur une phase.
I4	Explication de niveau 4: L'ouverture d'un disjoncteur ne peut se faire que sur une ou trois phases.
I5	Explication de niveau 5: L'ouverture sur une phase préserve l'état courant du réseau alors qu'une ouverture sur trois phases isole totalement l'élément avec la faute.

*Figure 9: Exemple de règle avec l'adjonction d'explications*

L'explication de niveau 1 est une explication minimale pour informer l'opérateur de l'existence d'un problème. Rien d'autre ne peut être dit au niveau de la protection de distance car ceci peut être la conséquence d'un problème ailleurs, sur un autre équipement. L'explication de niveau 2 donne le nom de la protection de distance et l'état des trois phases. D'autres informations de même type peuvent également être données (par exemple, l'état du disjoncteur associé). L'explication de niveau 3 donne la raison la plus usuelle alors qu'une autre raison est que la faute est sur une phase et la protection de distance s'est déclenchée de manière intempestive sur une phase. L'explication de niveau 4 correspond à la question classique "Pourquoi?" des autres systèmes experts. L'explication de niveau 5 fournit des connaissances stratégiques qui reflètent les choix au niveau de la compagnie qui veut que les perturbations du réseau entraînent le moins de désagrément possible pour les usagers. Une autre explication de même niveau est la "règle du n-1" de la compagnie: Entre une source d'énergie et le consommateur, il doit toujours y avoir un chemin de transmission d'énergie même si n-1 chemins étaient coupés simultanément (rappelons que la redondance joue un rôle important dans ce domaine). Ainsi l'opérateur peut associer une explication stratégique (niveau 5) avec des explications plus tactiques (niveaux 1 à 3) en définissant en début de diagnostic une interprétation (par exemple, I2 + I3).

Il est intéressant de noter que chaque explication fournit une information contextuelle qui n'est pas nécessaire pour le diagnostic, mais conditionne ce diagnostic. Il y a d'autres informations contextuelles qui sont gardées plus profondément implicites comme le fait que les équipements fonctionnent tant que la faute existe sur le réseau (les équipements arrêtent de fonctionner quand la faute disparaît). Cette approche laisse à l'opérateur un large choix d'explications, de l'explication sur la phase incriminée pour prendre sa décision à l'explication de niveau 5 suivant son expérience dans la compagnie. Par ailleurs, plus une information contextuelle est générale, plus elle concernera de règles. Cet exemple souligne bien l'interdépendance entre la génération de bonnes explications et l'utilisation explicite du contexte.

## **6. Le configurateur de connaissances KA7**

### **6.1. Origine de la conception de KA7**

Avec la réalisation du simulateur, et du couplage de celui-ci avec le diagnostiqueur au sein une même plate-forme, une forme de validation théorique du système global a pu être réalisée. Un incident donné, une topologie donnée, et une ou plusieurs défaillances d'équipement sont générées aléatoirement et fournis en entrée du simulateur. Le simulateur simule la suite d'événements qui surviennent et produit en sortie une séquence de signalisations telle qu'il en apparaît sur le consigneur d'état (il était même possible de décider de supprimer certaines signalisations qui apparaissaient au même instant qu'une autre, ou de manière arbitraire pour tester la robustesse du diagnostic au manque de faits initiaux (signalisations du consigneur d'état). Cette sortie du simulateur devient l'entrée du diagnostiqueur avec la donnée de la

topologie. Le diagnostiqueur déroule alors son expertise en deux étapes: un diagnostic rapide et son explication basé sur un modèle des événements. Il produit en sortie la faute qu'il a identifiée et les défaillances éventuelles d'équipement. Tout écart entre l'entrée du simulateur et de la sortie du diagnostiqueur est notifiée à l'opérateur à des fins d'analyse.

Outre cette validation théorique, EDF a procédé à une validation pratique sur site dans un poste à THT dans le nord de la région parisienne. La version de SEPT installée dans ce poste tournait en parallèle avec les opérations humaines habituelles dans le poste (SEPT était directement alimenté par les données se trouvant sur le consignateur d'état, et la topologie du poste au moment du problème était fournie par un autre logiciel). Les opérateurs de ce poste à THT contrôlaient ensuite le diagnostic de SEPT par rapport à leur diagnostic. Au bout d'une année, il a été possible de corriger quelques erreurs mineures dans SEPT, et SEPT a permis d'identifier des défaillances d'équipement qui n'avaient pas été détectées par les opérateurs.

Fort de ces deux validations, EDF a alors décidé de débiter une installation progressive de SEPT dans tous les postes à THT en France en commençant par un poste dans les Alpes qui présentait des caractéristiques très différentes du poste de la région parisienne. Il s'est rapidement avéré qu'il n'était pas question de porter SEPT sans des modifications si en profondeur qu'il s'agissait pratiquement d'une réécriture de SEPT. En effet, bien que les connaissances soient les mêmes dans tous les postes à THT, le contexte dans lequel étaient exploitées ces connaissances était très différent. Par exemple, en montagne il n'est pas possible d'espacer les équipements comme en plaine. Ceci avait, par exemple, des conséquences sur la manière dont les informations étaient rapatriées sur le consignateur d'état (les "retombées" d'information n'étaient pas enregistrées alors qu'une partie du diagnostic de SEPT en dépendait). D'autres raisons existaient aussi: seul le fonctionnement de certains équipements était enregistré, certains équipements non homologués par la compagnie étaient utilisés (et l'expertise ne se trouvait pas dans SEPT), etc.

## **6.2. La réalisation de KA7**

Afin de faciliter cette opération de réécriture de SEPT pour avoir une version adaptée aux spécificités de chaque poste à THT, il a été décidé de développer un système capable, partant des bases de connaissances de SEPT et des spécifications données par l'opérateur sur son poste à THT, de produire un "clone" de SEPT qui réponde aux exigences requises. Notre approche était d'autant plus justifiée que l'utilisateur du système n'était pas un novice, il connaissait très bien son poste à THT et n'était pas étranger à l'informatique.

Ce système a été appelé KA7. KA7 est un acronyme qui signifie "Knowledge Acquisition for SEPT". Le système KA7 vise à développer des clones de SEPT qui soient adaptés à des environnements spécifiques (topologies, matériel présent sur le site, adaptation des "règles EDF" au contexte courant, etc.), car bien que tous les postes à THT soient supposés effectuer la même tâche, ils se trouvent dans des contextes différents, l'un au bord de la mer, l'autre en haute montagne. Par conséquent, même si les postes à THT étaient supposés comporter les mêmes équipements, les contraintes environnementales imposaient une réécriture du système expert SEPT pour chaque poste. C'est pour rendre cette réécriture la plus automatique possible (et donc assurer la portabilité de SEPT) qu'il a été décidé de développer le système KA7 afin d'adapter le système expert SEPT aux conditions particulières d'un poste. KA7 agit comme un véritable configurateur de connaissances et génère des clones de SEPT qui correspondent aux spécifications qu'il acquiert auprès des exploitants. Comme effet de bord intéressant, cette approche fournissait les moyens d'assurer une certaine pérennité aux clones de SEPT qui pouvaient ainsi être facilement et rapidement adaptés aux évolutions technologiques. Une des clés de ce projet était la réutilisation de l'existant afin de ne pas repartir de zéro. Une autre clé était l'opérationnalisation effective et rapide du clone.



### **6.3. L'organisation de KA7**

Le système KA7 avait trois tâches principales à accomplir:

- l'acquisition interactive des connaissances pour gérer les bases de connaissances existantes,
- la configuration automatique des connaissances de SEPT pour un poste à THT spécifique,
- la certification pour vérifier que le comportement du clone produit correspondait aux spécifications initiales.

KA7 faisait appel à des formalismes de représentation où les connaissances acquises par KA7 gardaient toute l'expressivité requise pour SEPT, et où les passages d'un formalisme à un autre étaient assurés. Le choix du type de représentation était particulièrement important car le "langage" dans lequel était écrit les clones devait être compréhensible par des utilisateurs très différents. En cas de détection d'une limite de la conception, cette approche apportait une solution au problème de la correction de cette limite et sa propagation dans les clones déjà existant (génération d'un nouveau clone et d'une nouvelle documentation afférente).

KA7, en tant que configurateur, devait permettre de remplacer l'ingénieur de la connaissance en fournissant la possibilité à l'utilisateur de saisir directement ses connaissances dans un formalisme qui soit à la fois adéquat pour la tâche de diagnostic et immédiatement compréhensible par l'opérateur. KA7 a servi aussi à l'enrichissement des bases de connaissances en entrée de SEPT pour intégrer toute nouvelle technologie et expérience particulière rencontrée avec l'un de ses clones. La construction d'un système à base de connaissances par KA7 faisait donc appel à un ensemble de tâches à exécuter de manière coordonnée (simulation, apprentissage, acquisition explication, validation, etc.).

La représentation de connaissances hétérogènes exige l'appel à un langage de description de haut niveau. En entrée d'un configurateur de connaissances, est un ensemble de bases de connaissances (celles de SEPT), et l'ensemble des spécificités demandées pour un poste à THT particulier. Un des buts de KA7 était donc de rendre compatible la demande d'un utilisateur en fonction de ce dont il disposait déjà. En sortie, le configurateur de connaissances produit un système à base de connaissances, véritable clone de SEPT qui satisfaisait les exigences requises. Comme effet de bord, KA7 assurait aussi la bonne intégration du clone de SEPT dans l'environnement informatique du poste à THT concerné.

A un autre niveau, KA7 permettait l'enrichissement incrémental des bases de connaissances de SEPT. En effet, une nouvelle connaissance ne doit pas seulement être stockée dans une base de connaissances, mais aussi et surtout intégrée aux autres connaissances (établissement des liens entre elles). L'intérêt majeur de cette acquisition incrémentale des connaissances dans KA7 est que ces connaissances sont demandées quand elles sont nécessaires, et donc le système peut identifier (et enregistrer) clairement quel est leur contexte d'utilisation. En ce sens, le contexte d'utilisation des connaissances jouait un rôle essentiel dans KA7.

## **7. Bilan de l'expérience SEPT**

### **7.1. Exploiter l'organisation des connaissances**

La cohabitation de plusieurs tâches au sein d'un même système entraîne la nécessité de gérer différentes sources de connaissances, certaines sources correspondant à des connaissances "brutes" comme les connaissances du domaine, d'autres sources correspondant à une expression compilée de connaissances en provenance d'autres sources, comme les connaissances fonctionnelles ou le savoir faire qui est efficace pour une tâche donnée (Chandrasekaran, 1996).

Les connaissances du domaine se décomposent généralement en connaissances structurelles et en connaissances fonctionnelles. Les premières permettent de définir le squelette au SBC. La représentation des deuxièmes pose un problème. En effet, chaque tâche utilise généralement une version compilée de ces

connaissances fonctionnelles particulièrement efficace pour cette tâche. Il s'agit donc de pouvoir, partant d'une représentation directe de ces connaissances fonctionnelles pouvoir en faire une compilation automatique adaptée à chaque tâche, voire directement au moment de l'exécution et de leur utilisation. Il nous semble que ce problème doit être traité au niveau connaissance, pas au niveau implémentation (j'ai un bon formalisme, dites moi quel est votre problème).

### ***7.2. Exploiter les capacités de la machine pour raisonner***

Le savoir faire humain provient de la création d'heuristiques qui permettent à l'esprit humain d'aller plus rapidement à la solution malgré un nombre très grand de pistes à explorer. Vu le développement technologique actuel, les ordinateurs ont des capacités de stockage et de rapidité d'exploitation qui deviennent très supérieures aux capacités humaines correspondantes. L'exemple le plus marquant est le dernier match aux échecs entre Kasparov, le meilleur joueur d'échecs au monde et "Big Bue" l'ordinateur d'IBM conçu et réalisé dans le dessin de gagner cette partie d'échecs: l'ordinateur a gagné par sa puissance et sa rapidité d'exploitation des données. Bien que ce match ne remette pas en cause la suprématie de l'esprit humain sur la machine, la leçon est que le raisonnement de la machine ne doit pas nécessairement être un modèle fidèle du raisonnement humain. L'application SEPT avait déjà exploité cette constatation pour permettre à SEPT de développer un raisonnement de diagnostic qui contrôle de manière systématique et exhaustive l'ensemble des équipements. En fait, on ne fait ainsi que reconnaître une fois de plus qu'il peut exister plusieurs manières d'atteindre un but donné.

Par contre, il y a un prix à payer. En effet, plus le raisonnement de la machine sera différent du raisonnement humain, plus sa justification et son explication devront être soignées pour combler le fossé entre les deux types de raisonnement pour justifier ce que la machine a trouvé, et pour convaincre du bien fondé de sa démarche dans les termes humains. La présentation des informations et des résultats est un challenge d'autant plus important à relever que l'utilisateur qui nous intéresse est loin d'être un novice.

### ***7.3. Renforcer l'importance de la simulation***

La simulation est un outil particulièrement précieux pour justifier et convaincre en amont, et prédire en aval. Cet outil est ainsi utilisé couramment par l'homme aussi bien pour vérifier la démarche qui a été suivie que pour comparer des démarches alternatives avant d'en choisir une. La simulation permet d'avoir une vue cohérente sur un ensemble de données important et complexe. Un autre intérêt de la simulation, à notre avis, est que la complexité en question résulte bien souvent d'une considération de données, informations et connaissances hors de leur contexte. En fait, la simulation donne une représentation contextualisée du comportement d'un système qui se manifeste toujours dans un contexte d'occurrence précis. Dans le cas de l'application SEPT, ce contexte est généralement décrit par un ensemble de conditions initiales (la topologie du réseau électrique) et la description d'une faute sur ce réseau (temps d'apparition, nature de la faute, etc.).

La simulation présente d'autres intérêts comme l'aide à la formation qui permet de "rejouer" un incident dans différents contextes pour étudier le choix d'une méthode de résolution ou une stratégie sur une autre selon le contexte d'occurrence. Les opérateurs utilisent beaucoup ce moyen lors de leur prise de service, d'une part pour s'informer sur les incidents qui sont survenus et pour réfléchir sur le choix d'une stratégie qui a été appliquée par rapport à une autre, ceci afin de déterminer comment le choix d'une méthode ou d'une stratégie de résolution dépend du contexte d'occurrence.

### ***7.4. Expliciter le rôle du contexte***

Parmi les premières constatations faites en intelligence artificielle sur la notion de contexte (Brézillon, 1999), on peut retenir ici est que l'on ne peut parler de contexte qu'en relation avec un item: le contexte d'utilisation, le contexte de la situation, le contexte de la tâche, le contexte d'une activité, etc. Suivant la nature de l'objet en question, le contexte sera statique ou dynamique. Le contexte d'utilisation d'un

ensemble de connaissances peut être défini de manière statique, alors que le contexte d'une activité sera plus dynamique dans la mesure où l'activité transforme l'environnement, et donc partiellement le contexte.

Par rapport aux connaissances, plusieurs contextes sont à considérer comme le contexte d'acquisition d'une connaissance et le contexte de son utilisation, ces deux contextes pouvant d'ailleurs être un seul et même contexte dans une acquisition incrémentale de connaissances. Dans le cadre de l'application SEPT, un des rôles du contexte est de retrouver (compléter) des données qui n'ont pas été enregistrées bien que l'action correspondante ait été exécutée. Par exemple, l'existence du signal "ouverture du disjoncteur" implique que la protection de distance a envoyé auparavant un ordre de déclenchement au disjoncteur, même si le signal correspondant n'est pas enregistré. Cette information contextuelle nous est donnée par le modèle de bon fonctionnement d'un système de contrôle. Ce rôle du contexte est relié à une des définitions du contexte: "ce qui permet de ne pas tout dire dans une histoire".

Par rapport aux explications, il existe un lien très étroit entre contexte et explication. Tout d'abord, l'explicitation du contexte joue un rôle important dans la génération d'explications pertinentes. En effet, les explications font souvent appel à des connaissances contextuelles qui n'interviennent pas directement dans la résolution du problème mais contraignent cette résolution. Dans le cadre de l'application SEPT, l'explication de niveau 5 de la Figure 9 en est un exemple. Il faut noter également dans la même Figure 9 que la structuration des connaissances contextuelles à une étape de la résolution d'un problème (la métaphore de l'oignon dans Brézillon et al., 1997) permet de générer des explications à différents niveaux de détails.

Par rapport à une étape donnée du raisonnement, l'explicitation du contexte et de ses liens avec le raisonnement permet de dégager diverses vues sur le raisonnement: en amont sur la partie du raisonnement déjà tenu, à l'étape considéré par un modèle local, des préconditions et postconditions (voir la Figure 3 ci-dessus pour l'application SEPT), et en aval en permettant de déployer les alternatives et leurs poids relatifs. En amont, le contexte permet d'expliquer ce qui a été fait et est une justification. En aval, le contexte se présente comme un outil d'aide à la décision dans le choix de la meilleure alternative pour continuer de dérouler le raisonnement. Pour ceci, le rôle du contexte est surtout manifeste au travers de l'utilisation de la simulation.

Par rapport à la représentation des connaissances et du raisonnement tenu sur ces connaissances, le contexte se présente comme un moyen de gérer dynamiquement les connaissances à exploiter à une étape donnée du raisonnement, et permet de focaliser le raisonnement sur ce qui est important à cette étape. Dans l'application SEPT (voir l'exemple de règle donnée Figure 7), l'identification d'un contexte précis (reconnaissance du disjoncteur d'un système de contrôle donné) permet de sélectionner pour le moteur d'inférences les seuls règles pertinentes, et même de les tester dans le seul cadre d'un disjoncteur précis, permettant ainsi au moteur d'inférences de passer de la logique d'ordre 1 à une logique des propositions. À un autre niveau, l'explicitation du contexte permet à un système de gérer plusieurs tâches simultanément, puisque les contextes individuels des tâches sont rendu compatible, comme le montre l'atomisation des connaissances au niveau implémentation dans l'application SEPT (voir Figure 6). L'utilisation explicite du contexte permet également de gérer dynamiquement la méthode à utiliser pour accomplir une tâche, apportant ainsi une solution élégante au choix figé d'une méthode à la conception qui était proposé auparavant (voir par exemple Chandrasekaran et Johnson, 1993). Au niveau des agents constituant le système au niveau connaissance, le contexte joue un rôle important dans la définition de l'agent communication dont l'action principale sera alors de permettre la compatibilité des différents contextes personnels des autres agents et de l'utilisateur.

Il apparaît de cette analyse rapide du rôle du contexte dans l'application SEPT que le contexte doit être considéré comme un élément essentiel des futurs systèmes que nous appelons pour cela les systèmes d'assistance intelligents basés sur le contexte (SAIC).

### ***7.5. Architecture d'un système d'assistance intelligent en contexte***

Parmi les leçons que l'on peut tirer de l'expérience que nous avons eue avec l'application SEPT, plusieurs leçons ont une portée qui dépasse le cadre de l'application pour concerner la conception et le développement de tout système d'assistance intelligent basé sur le contexte (SAIC). Tout d'abord, un système d'assistance intelligent doit posséder, outre les moyens pour accomplir la tâche pour laquelle il est conçu (par exemple, le diagnostic comme dans l'application SEPT), trois fonctionnalités de base:

- La configuration des connaissances qui permet: d'adapter la base de connaissances d'un SAIC à un contexte précis, de mettre à jour cette base de connaissances pour intégrer, par exemple, de nouvelles technologies, d'acquérir de manière incrémentale de nouvelles connaissances avec leur contexte d'utilisation, et de doter ainsi un SAIC d'une véritable faculté d'apprentissage.
- La simulation qui permet de rendre compte de la manière dont sont exploitées ces connaissances par un raisonnement. Il est ainsi possible de mener une validation de la base de connaissances, d'étudier les conséquences des nouvelles connaissances introduites, de confronter des hypothèses alternatives, par exemple dans le choix d'une stratégie de résolution, ou, à l'inverse, de "rejouer" un raisonnement avec différents niveaux de détail pour bien en comprendre les mécanismes.
- La gestion des interactions, d'une part entre les agents (diagnostic, simulation, configuration, etc.), et, d'autre part, entre l'utilisateur et les agents logiciels afin que l'utilisateur ne se retrouve pas avec une surcharge cognitive en devant gérer ses interactions avec des agents aux langages très différents. Par ailleurs, certaines questions de l'utilisateur peuvent faire appel à une combinaison de réponses partielles en provenance des différents agents, et la réponse peut demander une coordination de ces éléments de réponse.

Chacune de ces fonctionnalités peut être considérée comme la tâche d'un agent (au sens des systèmes multi-agents). Un système d'assistance intelligent serait ainsi un système multi-agent constitué à minima d'un agent communication, d'un agent simulateur et d'un agent configurateur. L'architecture d'un système d'assistance intelligent basé sur le contexte est ouverte dans le sens où de nouveaux agents peuvent être ajoutés, ou le développement d'un agent arrêté s'il ne correspond plus à un objectif de la compagnie pour laquelle le système est réalisé, sans remettre pour cela en cause le reste du travail qui a été fait.

L'architecture du système d'assistance intelligent qui vient d'être présentée a été décrite au "niveau connaissance" dans le sens que lui avait donné Newell (1982, 1993), mais cette architecture doit également être considérée au niveau implémentation, les deux expressions de l'architecture d'un SAIC pouvant présenter des différences dans la forme. En effet, chaque agent logiciel accomplit une tâche, et pour cela utilise une large partie des connaissances du domaine soit directement soit sous une forme compilée. Une partie de ces connaissances se trouve, bien sûr, partagée par tous les agents. La solution qui était initialement préconisée dans la littérature (voir par exemple Chandrasekaran et Johnson, 1993) consistait à dupliquer les connaissances pour chaque tâche ou sous-tâche, et sous une forme qui était la plus efficace pour accomplir la tâche. Toutefois, cela pose, à notre avis, de sérieux problèmes de maintenance et de mise à jour. Par exemple, dans le cadre de l'application SEPT, les deux agents au niveau connaissances (le diagnostiqueur et le simulateur de la Figure 5) ont des connaissances communes. Certaines de ces connaissances se retrouvent à l'identique d'un agent à l'autre, comme les connaissances structurelles (par exemple la topologie du poste à THT), et d'autres connaissances sont exploitées sous des formes différentes (les connaissances fonctionnelles sont compilées de différentes manières selon les savoir faire mis en jeu). Pour éviter la duplication des connaissances communes au niveau implémentation, il suffit de distinguer clairement les différents types de connaissances comme nous l'avons fait dans l'application SEPT (comparer la Figure 6 à la Figure 5).

## 8. Bibliographie issue de l'application SEPT

- Bau D.-Y.**, "Explication basée sur un modèle dans les systèmes d'aide au diagnostic: Application au système à base de connaissances SEPT". Thèse d'Université 1652, Université d'Orsay, France, 1991.
- Bau D.-Y. and Brézillon P.**, "Explanation-based diagnosis with the use of a model in control systems". *Eleventh Expert System & Their Applications*. Avignon, France, 1991.
- Bau D.Y. and Brézillon P.**, "Model-based diagnosis of power station control systems: the SEPT experiment", *IEEE Expert*, 1992, p. 36-44.
- Brézillon P.**, "Notion d'interprétation dans le système expert SEPT". *Revue d'Intelligence Artificielle*, **3**(3), 1989, p. 7-18.
- Brézillon P.**, "Interpretation and rule packets in an expert system". *Expert Systems Theory & Applications (IASTED)*. Zurich, Acta Press, Anaheim, 1989, p.38-40.
- Brézillon P.**, "Interpretation and rule packet in expert systems. Application to the SEPT expert system". *Knowledge Based Computer Systems*. Ramani S, Chandrasekar R & Anjaneyulu KSR (Eds.). Narosa Publishing House, New Delhi, Bombay (India), 1989, p.78-87.
- Brézillon P.**, "Utilisation de la notion d'interprétation dans le système expert SEPT". *7e Congrès Reconnaissance de Formes et Intelligence Artificielle (RFIA89)*, AFCET. Tome 1, 1989, p.297-305.
- Brézillon P.**, "Interpretation and rule packet in expert systems. Application to the SEPT expert system". *Lecture Notes in Artificial Intelligence, N° 444, Knowledge Based Computer Systems*. S.Ramani R.Chandrasekar K.S.R.Anjaneyulu (Eds.). Springer-Verlag, 1990, p. 78-87.
- Brézillon P.**, "Explanations based on rule packets and interpretation in the METAL langage". *Proceedings of the 5th Workshop on Explanations*. Manchester, UK, 1990.
- Brézillon P.**, "Elaboration du système expert explicatif SEPT: III. Un diagnostiqueur en METAL pour la surveillance d'équipements dans un poste à Très Haute Tension". Rapport Interne 550, LRI, Université d'Orsay, France, 1990.
- Brézillon P.**, "METAL: a language for structured knowledge-based systems". *Proceedings of the IJCAI-91 Workshop on Software Engineering for Knowledge-Based Systems*. Sydney, Australia, 1991, p. 11-22.
- Brézillon P.**, "Explications utilisant la structuration d'un système à base de connaissances". *Actes de la 8ème Congrès Reconnaissance des Forme et Intelligence Artificielle (RFIA-91)*, Vol. III, 1991, p. 1167-1176.
- Brézillon P.**, "Les explications et les systèmes à base de connaissances". Rapport de Recherche N° 21/91, LAFORIA, Université Paris VI, Paris, France, 1991.
- Brézillon P.**, "METAL: a language for structured knowledge-based systems". Rapport de Recherche 23/91, LAFORIA, Université Paris VI, Paris, France, 1991.
- Brézillon P.**, "Le configurateur de connaissances KA7: Synthèse bibliographique". Rapport de Recherche 22/91, LAFORIA, Université Paris VI, Paris, France, 1991.
- Brézillon P.**, "SEPT: a diagnoseur with a simulator". *Proceedings of the 2nd Expert Systems and Computer Simulation in Energy Engineering*. Erlangen, Germany, 1992.
- Brézillon P.**, "Building explanation during expert-computer interaction". *Proceedings of the East-West Human-Computer Interaction*. St Petersburg, CEI, Part II, 1992, p. 378-385.
- Brézillon P.**, "Architectural and contextual factors in explanation construction". *Proceedings of the ECAI'92 Workshop on Improving the Use of Knowledge-Based Systems with Explanations*. Vienna, Austria. Research Report 92/21, LAFORIA, University Paris VI, France, 1992, p. 65-74.
- Brézillon P.**, "Proceedings of the ECAI-92 Workshop W15 on Improving the Use of Knowledge-Based Systems with Explanations." Rapport de Recherche 92/21, LAFORIA, Université Paris VI, Paris, France, 1992.
- Brézillon P.**, "Building explanation during expert-computer interaction". Rapport de Recherche 92/26, LAFORIA, Université Paris VI, Paris, France, 1992.

- Brézillon P.**, "Contextualized explanations". Rapport de Recherche 92/28, LAFORIA, Université Paris VI, Paris, France, October 1992.
- Brézillon P. and Bau D.-Y.**, "Elaboration du système expert explicatif SEPT: choix du couplage entre le simulateur et le diagnostiqueur". Rapport Interne n° 508, LRI, Université d'Orsay, France, 1989.
- Brézillon P. and Bau D.-Y.**, "Elaboration du système expert explicatif SEPT: II. Un simulateur de poste à Très Haute Tension en METAL". Rapport Interne 548, LRI, Université d'Orsay, France, 1990.
- Brézillon P. and Bau D.-Y.**, "Diagnostic basé sur un modèle des systèmes de contrôle dans les postes des réseaux d'énergie: l'expérience SEPT". *Revue d'Intelligence Artificielle*, **6(4)**, 1993, p. 407-430.
- Brézillon P. and Fauquembergue P.**, "Surveillance des dispositifs de protection dans un poste à T.H.T. et localisation des défauts par analyse des informations fournies par le consignateur d'état. Présentation des travaux en cours". Note EDF HR.32-0847, HR.47-0264, 1987.
- Brézillon P. and Fauquembergue P.**, "Surveillance des dispositifs de protection et de contrôle commande dans un poste à T.H.T. par une approche système expert: bilan de l'étude de faisabilité". Note EDF HR.34-0908, 1987.
- Brézillon P. and Fauquembergue P.**, "SEPT : un système expert pour la Surveillance d'Equipements dans un Poste à très haute Tension". EDF, *Bulletin de la Direction des Etudes et Recherches*, Série B, Réseaux Electriques, Matériels Electriques, n° 3, 1988, p. 5-10.
- Brézillon P. and Fauquembergue P.**, "Development of an expert system for a synthesis of events occurring in an EHV substation". *Proceedings of the 12th IMACS*. Paris, France, 1988.
- Brézillon P. and Fauquembergue P.**, "La tâche de diagnostic dans le système expert SEPT". *Proceedings of the Tenth Expert Systems and Their Applications*. Avignon, France, 1990, p. 283-293.
- Brézillon P., Fauquembergue P. and Hertz A.**, "SEPT, an expert system approach for the monitoring of EVH substation control equipment". *Proceedings of the Expert Systems Application to Power System*. Stockholm-Helsinki. 1988, p. 6.9-6.13.
- Brézillon P., Bau D.-Y., Fauquembergue P., Hertz A. and Maizener A.**, "Elaboration of the SEPT expert system as a coupling of a simulator and a diagnostician". *Third Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. Charleston, South Carolina, USA, Vol. I, 1990, p. 54-60.
- Fauquembergue P. and Brézillon P.**, "SEPT : maquette de système expert pour surveiller le matériel de contrôle-commande d'un poste à THT". D.E.R.87 faits marquants, EDF-DER (Ed.), 1987, p. 83-84.
- Fauquembergue P. and Brézillon P.**, "Vers une utilisation de systèmes experts pour surveiller les équipements des postes électriques à très haute tension". *Revue Générale d'Electricité*, n° 10, 1988, p. 36-37.
- Fauquembergue P. & Brézillon P.**, "SEPT: un système expert pour la surveillance d'équipements dans un poste à très haute tension". Note EDF HR.47.601, HR.34.1073, 1988.
- Fauquembergue P., Brézillon P. and Harmand Y.**, "Synthesis of events in an EHV substation : an expert system approach". *IFAC Power Systems, Modelling and Control, Applications*. Brussels, 1988, p. 16.6.1- 16.6.5.
- Fauquembergue P., Brézillon P. and Harmand Y.**, "Surveillance d'équipements de protection et de contrôle-commande dans un poste à très haute tension: une approche système expert". Note EDF HR.47.0497, 1988.
- Perrot L.**, "Apport à la configuration de connaissances par réingénierie de systèmes existants. KA7: un système appliqué au domaine des réseaux électriques". Thèse d'Université TH96/07, Université Paris 6, France, 1996.
- Perrot L. and Brézillon P.**, "Configurateur de connaissances KA7 - Etat de l'art". Rapport Technique EDF, 153 pages, 1992.
- Perrot L., Fauquembergue P. and Brézillon P.**, "Projet KA7: Spécifications fonctionnelles du prototype". Rapport Technique EDF, 24 pages, 1992.
- Perrot L., Fauquembergue P. and Brézillon P.**, "Towards automatic generation of knowledge bases for diagnosis systems in the field of power systems". *4th Expert System Applications to Power Systems (ESAP'93)*. La Trobe University, Melbourne, Australia, 1993, p. 373-378.

**Perrot L, Taché O. and Brézillon P.**, "Knowledge acquisition, and configuration for electrical power systems". *5th International Conference on Intelligent System Applications to Power Systems (ISAP-94)*. Vol. 2, 1994, p. 487-494.

## 9. Bibliographie citée

**Brézillon, P.**, "Context in human-machine problem solving: A survey". *The Knowledge Engineering Review*, 14,1999, p.1-34.

**Brézillon, P., Gentile, C., Saker, I., and Secron, M.**, "SART: A system for supporting operators with contextual knowledge". *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)*. Federal University of Rio de Janeiro Ed., 209-222. 1997 (also available at <http://www-poleia.lip6.fr/~brezil/Pages2/CONTEXT-97/index.html> ).

**Chandrasekaran B. and Johnson T.R.**, "Generic tasks and task structures: History, Critique and new directions", In: *Second Generation Expert Systems*, Springer Verlag, 1993.

**Clancey W.J.**, "Model construction operators". *Artificial Intelligence*, 53, 1992, p. 1-115.

**Mulet-Marquis, D. and Gondran, M.**, "Un langage pour les systèmes experts: ALOUETTE", *Bulletin de la Direction des Etudes et Recherches d'EDF*, Série C(3/4), 1985.

**Newell A.** (1982) "The knowledge level", *Artificial Intelligence*, 18, 1982, p.87-127.

**Newell A.**, "Reflections on the knowledge level", *Artificial Intelligence*, 59, 1993, p. 31-38.

Patrick Brézillon est chercheur au CNRS après avoir obtenu sa Thèse d'Etat en 1983 à l'Université Pierre et Marie Curie. Le sujet de sa thèse était la la modélisation mathématique des métabolismes calcique et de la sérotonine, le premier sous forme d'un modèle nonlinéaire auto-oscillant. Depuis, ses préoccupations sont à la frontière des approches modélisation mathématique et intelligence artificielle autour du développement des systèmes d'assistance intelligents basé sur le contexte, avec la mise au point d'un formalisme de représentation basée sur le contexte des connaissances et du raisonnement appelé les graphes contextuels. Il est l'un des fondateurs de la communauté s'intéressant au contexte. Sa page web est à <http://www-poleia.lip6.fr/~brezil/>