

L'IA au service de l'engagement des secours

Guy A. Narboni¹, Commandant Nicolas Mathieu²

¹ implexe - ToMCo

² SDIS de Seine-et-Marne - Agence du Numérique de la Sécurité Civile

r-d@implexe.eu

Résumé

Cette présentation du moteur de mobilisation développé pour le projet NexSIS décortique la mécanique d'engagement des secours qui fait suite à un appel d'urgence aux pompiers. En suivant le fil d'une modélisation en Prolog, elle illustre le rôle toujours actuel de l'« IA symbolique » dans une conception sur mesure de systèmes d'aide à la décision, au plus proche du métier.

Mots-clés

Automatisation du raisonnement, Modélisation par Contraintes, Programmation en logique, Programmation mathématique, Systèmes d'aide à la décision.

Abstract

This overview of the 'mobilisation engine' developed for the French NexSIS project details the mechanics of the dispatch process that follows an emergency call for Fire & Rescue Services. Using Prolog modeling as guiding thread, it illustrates the long-lasting contribution of 'Symbolic AI' in the design and fine-tuning of custom-built Decision Support Systems.

Keywords

Automation of reasoning, Constraint-based modeling, Logic programming, Mathematical programming, Decision-Support Systems.

1 Introduction

Porté par l'Agence du numérique de la sécurité civile, « NexSIS 18-112 » est un grand projet national [1] d'unification des systèmes d'information et de commandement des services d'incendie et de secours (IS). Son objectif est d'améliorer la gestion des interventions d'urgence à travers une nouvelle génération de Systèmes de Gestion « souverains » dont les principaux modules sont dédiés :

- à la cartographie (SIG)
- à la réception et à la qualification des Appels d'urgence (SGA)
- à la création et au suivi des Opérations (SGO)
- aux Echanges d'informations (SGE) avec les autres partenaires de l'urgence comme le SAMU ou les forces de l'ordre.

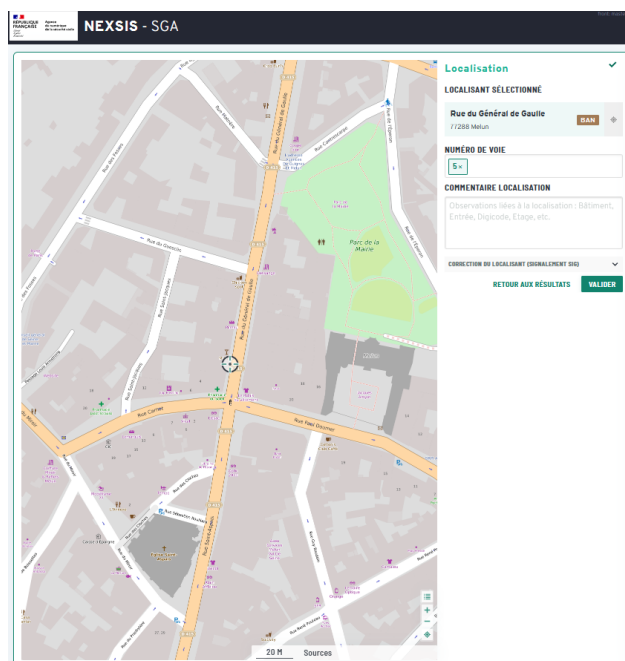


FIGURE 1 – Alerte notifiée par le SGA



FIGURE 2 – Prise en compte par le SGO

La création d'une opération est la réponse apportée par le Centre Opérationnel Départemental d'Incendie et de Secours (CODIS) à une situation d'urgence, localisée sur son territoire et concernant la protection des personnes, des biens et de l'environnement.

Les services territoriaux d'IS disposent sur leur secteur de moyens d'intervention dont la répartition géographique découle d'une analyse de risques et d'une stratégie de « couverture opérationnelle » de ces derniers.

Traiter une alerte, c'est répondre vite et bien à l'urgence en envoyant rapidement sur les lieux de l'incident les secours appropriés. C'est en deux mots mobiliser les bonnes ressources, au bon endroit et au bon moment.

Le « moteur de mobilisation » est la pièce maîtresse de NexSIS dans ce processus. Il doit aider l'opérateur dans l'évaluation de la situation présentée et dans la prise de décision d'engagement des secours.

L'apport principal du moteur concerne la phase « réflexe » de la mobilisation où il s'agit, en temps réel, de planifier les mesures à prendre en réaction immédiate, de manière systématique, avec des moyens structurants (dimensionnés sur la base de scénarios « majorants »). Par exemple : un signalement de feu ordinaire déclenchera une mission de lutte contre l'incendie qui se traduira réglementairement par un départ de fourgon pompe-tonne avec 6 pompiers à bord.

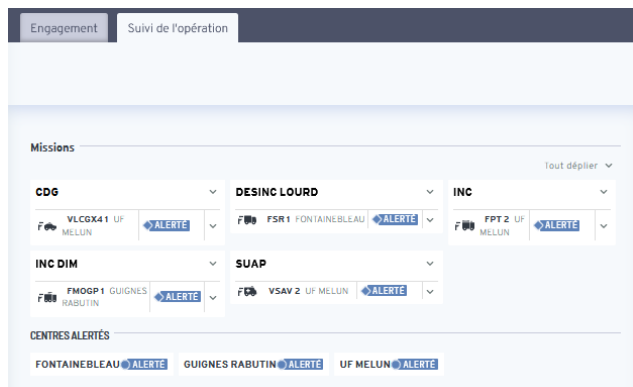


FIGURE 3 – Opération engagée



FIGURE 4 – Vue détaillée de l'équipage du fourgon INC



FIGURE 5 – Ordre de départ des moyens

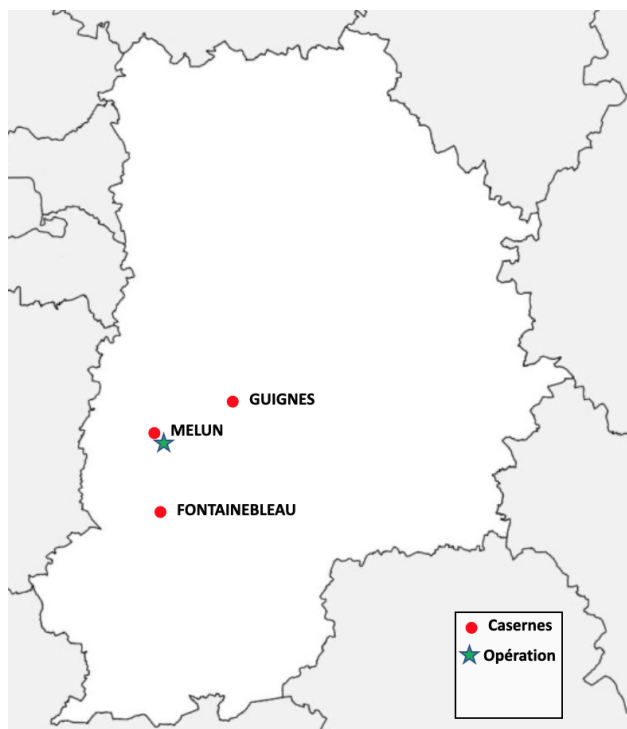


FIGURE 6 – Casernes mobilisées pour l'opération

La réponse du moteur se construit en deux temps [2][3][4] :

1. définition d'une réponse tactique — en termes de « missions » à accomplir (c'est-à-dire, en vue d'effets à obtenir)
2. définition sur cette base d'une réponse opérationnelle — par une affectation concrète de ressources (matérielles et humaines) aux missions.

Cette « traduction » de la situation d'urgence en moyens à engager est un challenge technique où, nous allons le voir, l'IA a toute sa part.

2 Le moteur de planification et la réponse tactique

La situation remontée exige-t-elle d'intervenir ou non ? La décision d'intervention est du ressort du service d'IS territorialement compétent. S'il y a intervention, il faut la planifier comme une opération.

La mécanisation de ce processus de décision fait appel à un moteur de règles.

2.1 Adaptabilité au territoire

Au niveau national, la plate-forme NexSIS fournit un moteur générique. Pour fonctionner, ce système doit être complété par une base de connaissances spécifique qui formalise les pratiques opérationnelles propres au département concerné.

Le découpage thématique de cette base en grandes familles (feux, accidents, secours d'urgence à personne, etc.) ramène l'examen d'une situation d'urgence à l'application d'un jeu de règles plus limité qui précise la doctrine d'engagement dans ce cas.

2.2 Règles métier

Les règles d'engagement sont définies dans leur grandes lignes par le Règlement Opérationnel local. Elles s'énoncent en pratique sous la forme : « si la situation d'urgence présente telles et telles caractéristiques, alors il faut telle mission pour y faire face ».

Les conclusions des règles font intervenir non seulement la qualité de la réponse (le type de mission à effectuer) mais aussi sa quantité (le nombre de missions étant un indicateur du « volume » de secours à engager).

Les conditions des règles portent sur les attributs qui qualifient la situation comme dans l'exemple de la figure 7, mais pas seulement. Elles peuvent également faire référence aux conclusions d'autres règles. En particulier, une condition peut porter sur le nombre total de missions, comme dans la figure 8 où la règle se lit : « si le nombre de missions engagées est supérieur à 3, alors il faut prévoir *a minima* une mission de commandement ».

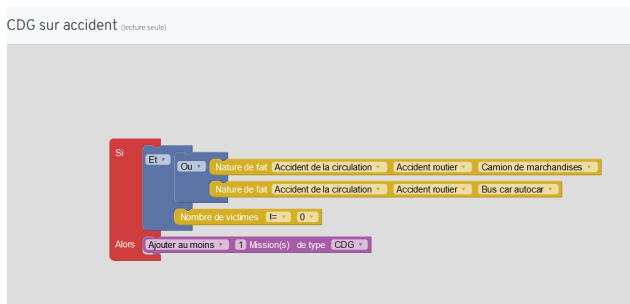


FIGURE 7 – Vue de la structure logique d'une règle

2.3 Modélisation

Les règles d'engagement doivent être pensées dans une logique monotone de montée en puissance de l'engagement : « plus j'ai ceci, plus je dois avoir cela ». En optimisation

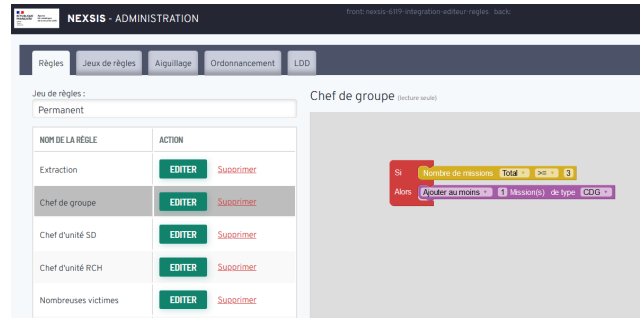


FIGURE 8 – Règle dépendant des résultats d'autres règles

multi-critères, on retrouve des notions similaires sous l'appellation de « règles de décision graduées » [5].

Ces règles s'éloignent du cadre normatif [6] retenu par la plupart des systèmes de gestion de règles métier. Nous avons donc conçu un « mini langage » déclaratif spécialement dédié à l'expression de ces connaissances [7]. Il est essentiellement de niveau propositionnel, ou plus exactement « pseudo-booléen » puisqu'il doit permettre de sommer des volumes.

Soit M la taille du catalogue de missions. Nous avons au plus M variables de décision (x_1, \dots, x_M) représentant des entiers positifs ou nuls qui correspondent au nombre des différents types de missions à engager.

Chaque règle est une implication logique : *conditions* \rightarrow *conclusion* où la conclusion impose à l'une des variables une valeur entière minimale $(x_j \geq c_j)$ et où chacune des conditions fait également intervenir un seuil. Ce seuil de déclenchement peut porter :

- soit sur une fonction booléenne y des paramètres d'entrée, à savoir les attributs booléens qui qualifient l'affaire (selon une nomenclature [8][9] désormais commune à tous les services d'urgence). Dans ce cas, il est égal à 1 :

$$y \geq 1$$

- soit sur une variable de décision :

$$x_k \geq c_k$$

- soit sur le nombre total des missions :

$$x_1 + \dots + x_M \geq c_k$$

Une fois la situation connue, les paramètres d'entrée s'effacent. Les règles qui ne s'appliquent pas sont tout simplement ignorées. Si bien que, techniquement, les règles restant à satisfaire sont des disjonctions d'inégalités linéaires que l'on peut écrire sous la forme :

$$(\vec{a}_{1,r} \vec{x} \geq c_{1,r}) \wedge \dots \wedge (\vec{a}_{n,r} \vec{x} \geq c_{n,r}) \Rightarrow (x_r \geq c_r) \quad (R_r)$$

où \vec{x} désigne le m -uplet des variables de décision et où les coefficients des produits cartésiens sont positifs ou nuls (puisque'ils sont soit tous égaux à 1, soit tous égaux à 0 à l'exception d'un seul 1).

Cette syntaxe particulière généralise les clauses de Horn propositionnelles [10]. Il est à noter que si les inégalités

n'étaient pas toutes dans le même sens (dans le membre gauche de la règle), on pourrait poser avec ce langage n'importe quel problème SAT en variables binaires, et donc encourir le risque d'une explosion combinatoire dans la recherche d'une solution.

2.4 Résolution

Sur le plan technique, l'automatisation de la déduction repose sur la traduction des règles à satisfaire en un système de contraintes.

Le problème de la décision tactique se présente en fin de compte comme un problème d'optimisation sous contraintes : trouver le plus petit volume de secours satisfaisant à l'ensemble des règles énoncées. Autrement dit :

Minimiser f
sous les contraintes (R_r) du jeu de règles
et de la fonction de coût $f \geq x_1 + \dots + x_M$
avec les $x_i \geq 0$ et entiers.

Or, de la même façon que la résolution unitaire est complète pour les formules Horn SAT, on montre que la propagation est complète avec les jeux de règles exhibant cette structure [10]. Le résultat découle de l'observation que toutes les contraintes du système sont min-closes [11]. Le problème est donc décidable en temps polynomial.

Avec un solveur CLP(FD), la décision s'effectue sans avoir à recourir à l'énumération : si le produit cartésien des domaines des variables ne se réduit pas au vide, son plus petit élément est la solution optimale recherchée.

L'implantation du moteur se résume donc à l'écriture de quelques lignes de code. Avec des variables entières, la traduction des règles se fait naturellement¹. Et pour résoudre un cas, une seule inférence de la machine Prolog suffit.

2.5 Innovation

Le langage proposé a été validé sur une base représentative de cas d'usage, constituée au départ de 200 règles métier à conclusions multiples (soit l'équivalent d'un millier de clauses à conclusion unique ou proprement « Horn »).

Ses premiers utilisateurs cobayes semblent s'approprier sans trop de difficulté sa logique combinatoire où il n'y a pas d'ordre entre les règles et où toutes les règles s'appliquent de concert.

Comment le montrent les précédentes captures d'écrans, la rédaction de ces règles logiques par un « paramètre » s'effectue au travers d'une interface graphique, sans recourir à l'écriture d'une quelconque formule mathématique.

Avec une version libre de Prolog, les temps de réponse du moteur de règles sont de l'ordre du centième de seconde sur une machine standard, autrement dit, sans latence pour une application déployée d'emblée « dans le Cloud »².

L'innovation consiste ici à remplacer par des réponses métier beaucoup plus affinées et contextualisées un catalogue

1. Une forme propositionnelle serait nettement moins intuitive et l'on s'épargne aussi l'obstacle de la traduction des contraintes de sommation [12]

2. Sur la plate-forme NexSIS, le moteur Prolog est virtualisé dans un conteneur Docker.

de réponses codifiées qui ne couvre pas toutes les situations d'urgence — seulement les cas les plus courants et, qui plus est, avec des volumes prédéfinis.

A l'aide d'un petit nombre de règles, on peut en effet couvrir un très large éventail de situations. A travers ces règles, toutes les informations communiquées sur l'incident sont susceptibles de moduler la réponse opérationnelle : aussi bien la description structurée de l'alerte transmise par le SGA que son enrichissement automatique par le Système d'Information Géographique (notamment sur les risques propres à la zone d'intervention).

Mais l'innovation doit aller de pair avec la conduite du changement. Jusqu'où la doctrine doit-elle être poussée ? La contrepartie en effet est que la combinatoire des configurations de départ va grossir très vite. On ne pourra plus raisonner en « extension ». Il faudra raisonner en « compréhension ». Nous avons rappelé pour cela l'importance accordée à l'intelligibilité des jeux de règles. Leur cycle de vie devra être soumis à un processus de validation de cohérence rigoureux.

3 Le moteur d'affectation et la réponse opérationnelle

Une fois la réponse tactique définie, il s'agit de trouver dans les centres d'IS les ressources correspondantes pour lancer l'opération. Il faut pouvoir détacher les moyens prévus pour les dépêcher sur place. Est-ce possible au regard de l'inventaire des matériels et personnels disponibles ? Et si oui, comment assurer au mieux :

- la rapidité d'intervention
- la connaissance du terrain
- la qualité de service (quitte à la dégrader si l'on ne peut faire autrement)
- la capacité de répondre ultérieurement à une urgence potentiellement plus grave (par une préservation de la couverture opérationnelle des risques) ?

L'aide à la décision conduit clairement ici à la formulation d'un problème d'optimisation multi-critères. Sa mécanisation fait appel à un solveur de contraintes.

3.1 Déclinaison préalable

Que demande une mission ? Essentiellement un engin avec son armement au complet, constitué notamment de personnel formé. Là encore, la doctrine de la tutelle départementale s'applique. Elle s'exprime en Prolog par des règles qui indiquent quel engin retenir de façon préférentielle pour chaque mission. Elle précise également le nombre de postes à pourvoir, avec pour chacun d'eux les compétences idoines :

- compétences liées à la mission
- compétences liées à l'engin pour certaines manipulations
- compétences liées à son acheminement (typiquement, un permis pour le conducteur du véhicule).

Cette déclinaison donne à l'objectif fonctionnel une structuration organique adaptée au territoire.

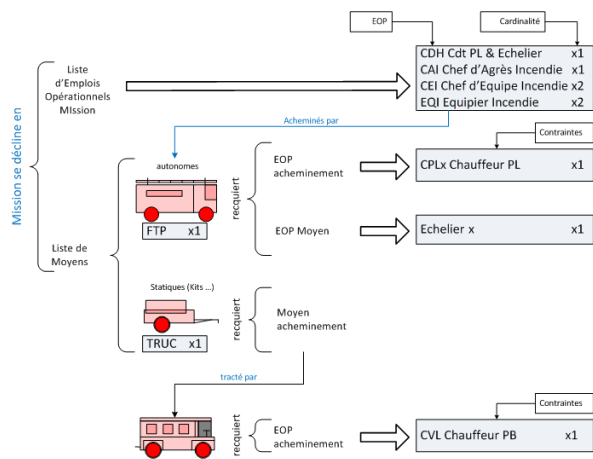


FIGURE 9 – Illustration de la déclinaison d’une mission (source : Document support au design SGO, ANSC, 2019)

3.2 Modélisation

La déclinaison des missions débouche sur la construction d’un modèle d’allocation des ressources en matériel et en personnel. Les casernes étant les centres de ressources, le problème principal consiste à affecter chacune des missions à une caserne.

Classiquement (cf. graphe de la figure 10), cela se modélise comme un problème « de transport » mettant en relation l’offre (les casernes) et la demande (les missions). Bien entendu, il faut que la caserne dispose de l’engin approprié avec un équipage mobilisable.

La résolution du problème principal dépend donc de la résolution de deux problèmes subordonnés :

- l’un relatif au choix des engins (nouveau problème de transport)
- l’autre, au choix des agents (problème d’affectation ou à nouveau de transport, selon que l’on individualise ou non les ressources en personnel).

Si l’on met tout cela à plat, des contraintes dites de liaison apparaissent entre ces 3 problèmes qui prennent la forme de contraintes d’intégrité : pour une mission m se décomposant en un besoin d’engin e , il ne peut y avoir d’affectation de e à un véhicule v de la caserne c que si c est affectée à m . Autrement dit, l’affectation de e à v implique l’affectation de m à c . Avec des variables binaires explicitant ces choix, cette implication se traduit par l’inégalité : $b_{e,v} \leq b_{m,c}$.

Il en va de même pour les besoins de postes.

Le système de contraintes à satisfaire pour que l’opération soit réalisable est donc formé par la réunion de 5 sous-systèmes :

- les contraintes du problème principal
- celles des deux problèmes subordonnés
- les contraintes des deux systèmes de liaison.

Le modèle peut être formulé avec des contraintes linéaires en Programmation en Nombres Entiers ou avec des contraintes globales en Programmation par Contraintes — option retenue ici pour rester dans l’univers CLP(FD).

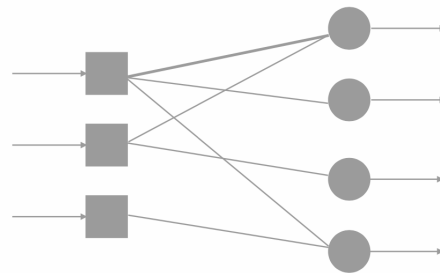


FIGURE 10 – Modélisation de la satisfaction des besoins (à gauche) par des ressources (à droite) en capacités limitées

Sur le plan « graphique », la capacité des arcs dans les problèmes subordonnés (graphes bipartis engins-véhicules et postes-agents) est conditionnée par la valeur des variables de décision du problème principal (les arcs missions-casernes devenant des coefficients annulateurs s’ils ne sont pas sélectionnés). Du point de vue de la complexité malheureusement, le problème résultant n’est plus un simple problème de flot.

3.3 Définition des objectifs

Si les contraintes à respecter se conçoivent aisément, il n’en va pas de même pour les critères d’optimisation. Prenons celui de la célérité : lorsque le dispositif se limite à une mission, la réponse la plus rapide vient de la caserne la plus proche. Mais si le dispositif est constitué de plusieurs missions, comment comparer deux réponses ? Une solution peut être plus rapide pour une mission et plus lente pour une autre. Faut-il prendre le min, le max, la moyenne des temps d’intervention ? La réponse n’est pas évidente.

La modélisation en contraintes proposée ouvre des perspectives auxquelles le métier ne sait pas encore répondre. Elle n’interdit pas de reproduire les heuristiques gloutonnes qu’emploient les logiciels de gestion actuels. Ces derniers effectuent une recherche séquentielle des moyens de secours selon un ordonnancement des casernes préalablement défini sur chaque portion de territoire. Mais ils ne peuvent produire ainsi qu’une seule proposition d’engagement.

Cette sorte d’optimisation « au niveau de la mission » a des effets pervers bien connus. Pour éviter de se voir préempter les compétences rares, elle incite à placer en premier les missions qui les exigent. Ceci a pour effet de dénaturer l’ordre métier qui ne correspond plus à la séquence d’arrivée des engins attendus sur les lieux de l’urgence.

En permettant de réserver à l’avance un volant de compétences rares, la pose des contraintes nous fait avancer vers une optimisation « au niveau de l’ensemble des missions ». En termes de priorité opérationnelle, la modélisation actuelle permet déjà de faire cohabiter les « listes de défense » (pour la connaissance du terrain) avec les « temps de transit » (pour la célérité), ce qui en soi est une nouveauté.

A l’avenir, la prise en compte d’autres axes comme la qualité et la préservation de la couverture ira dans le sens d’une optimisation plus globale.

3.4 Sélection finale

Dire qu'une mission peut être affectée à une caserne, c'est dire que cette caserne est en capacité d'armer un engin pour l'assurer. Or, sur le plan des ressources humaines, on trouve souvent des solutions équivalentes du point de vue de l'optimalité. C'est donc à ce moment qu'il convient de tenir compte des multiples règles en vigueur pour « prioriser » l'affectation des personnels dans les casernes en fonction :

- des plannings journaliers des feuilles de garde,
- du cumul des temps d'engagement,
- de l'ancienneté,
- ...

On touche ici à des questions d'organisation de l'activité dans les casernes qui sont au cœur de l'acceptabilité sociale de la solution proposée. Là encore, Prolog permet d'exprimer avec une relative aisance les heuristiques métier de sélection des agents.

3.5 Innovation

A défaut de savoir définir « la » meilleure solution, on pourra, en faisant varier les critères d'optimisation, proposer à l'opérateur du CODIS un bouquet de solutions, toutes correctes par construction. D'un côté, le système d'aide à la décision le libèrera de tâches répétitives qui ont peu de valeur ajoutée. De l'autre, il lui demandera une plus grande acuité de jugement dans le choix de l'ordre de départ. Loin de le « robotiser », l'application devrait au contraire le faire monter en expertise.

4 Conclusion

Nous avons présenté dans cet article les rouages essentiels de la mécanique d'engagement des secours. La réponse « théorique » définit en quelque sorte le cadre de l'opération et la réponse « instanciée » complète ce tableau.

Pour les pompiers, le moteur de mobilisation est un outil de logistique opérationnelle. A l'image d'un système expert, il est par conception « paramétrable » par des doctrines, afin d'apporter des réponses respectueuses des spécificités locales.

Pour le développement et le déploiement de ce composant à forte valeur ajoutée, l'Agence du numérique de la sécurité civile a fait le pari de Prolog. L'une des raisons est que ce langage permet de « traduire le métier » de manière visible et compréhensible, et par là de modéliser beaucoup plus finement les besoins. Nous espérons avoir montré qu'il se prêtait aussi admirablement à la résolution des problèmes posés.

Remerciements

A l'Agence, à nos collègues de la BSPP, du SDMIS, des sociétés ToMCo et Cosytec, sans qui ce beau projet ne se serait pas devenu réalité.

A Alain Colmerauer, pour avoir inventé Prolog, il y a tout juste 50 ans.

Références

- [1] Préfet Guillaume Lambert. Création d'un système unifié de traitement des appels, des alertes et des opérations des services d'incendie et de secours. Ministère de l'Intérieur, 2016.
- [2] J. Rohmer and M. Carruel. Moteurs de règles et de contraintes dans NexSIS : Besoins et recommandations. Note d'orientation, ToMCo, 2020.
- [3] Agence du Numérique de la Sécurité Civile. Rapport d'activité, 2020.
- [4] N. Mathieu and N. Mortada. Moteur de mobilisation. Présentation au 127 congrès des Pompiers de France, 2021.
- [5] D. Dubois and H. Prade. Towards a reconciliation between reasoning and learning - a position paper. In *Procs. SUM 2019*, LNAI 11940, 2019.
- [6] Object Management Group. Decision model and notation. <https://www.omg.org/spec/DMN>, 2015.
- [7] G. Narboni. From deontic logic to mathematical optimization : a rule formalism for emergency decision-making (extended abstract). In *Procs. Int. Conf. on the Integration of Constraint Programming, Artificial Intelligence and Operations Research*, CPAIOR 22, Los Angeles, 2022.
- [8] OASIS Emergency Management TC. Emergency data exchange language (EDXL). Organization for the Advancement of Structured Information Standards, 2009.
- [9] GT CISU. Cadre d'interopérabilité des services d'urgence. Groupe de travail interministériel, 2019.
- [10] G. Narboni. On rule systems whose consistency can be locally maintained. *AI Communications*, 26, 2013.
- [11] P. Jeavons and M. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2), 1995.
- [12] O. Bailleux and Y. Boufkhad. Efficient CNF encoding of boolean cardinality constraints. In *Procs. Int. Conf. on Principles and Practice of Constraint Programming*, CP 03, LNCS 2933, 2003.