

# Plateforme et ontologie de scénarios pour l'aide à la veille stratégique en source ouverte

Emilien BONDU(\*,\*\*), Stephan BRUNESSAUX(\*), Habib ABDULRAB(\*\*), Nathalie CHAIGNAUD(\*\*), Jean-Philippe KOTOWICZ(\*\*)  
[emilien.bondu@litislab.eu](mailto:emilien.bondu@litislab.eu), [stephan.brunessaux@eads.com](mailto:stephan.brunessaux@eads.com), [habib.abdulrab@insa-rouen.fr](mailto:habib.abdulrab@insa-rouen.fr), [nathalie.chaignaud@insa-rouen.fr](mailto:nathalie.chaignaud@insa-rouen.fr), [jean-philippe.kotowicz@insa-rouen.fr](mailto:jean-philippe.kotowicz@insa-rouen.fr)

(\* ) [EADS Defence and Security Systems](#), Parc d'Affaire des Portes 27106 Val de Reuil (France),

(\*\*) [LITIS](#), Avenue de l'Université - BP 8 - 76801 Saint-Étienne-du-Rouvray Cedex (France)

## Mots clefs :

veille stratégique, sources ouvertes, plateforme d'intégration, gestion des connaissances, ontologie, signal faible, scénarios

## Keywords:

business intelligence, open source, integration platform, knowledge management, ontology, weak signal, scenarios

## Palabras clave :

Inteligencia empresarial, fuente abierta, plataforma de la integración, gestión del conocimiento, ontología, señal débil, escenarios

## Résumé

L'activité de veille stratégique sur des sources ouvertes d'information est aujourd'hui de plus en plus complexe. En effet, avec la croissance du web 2.0 et la multiplication des gisements d'informations, le veilleur se trouve confronté, d'une part, à une masse de données toujours plus importante et d'autre part, à une quantité de formats et structures de données croissante. Dans ce contexte, la détection de *signaux faibles*[1] par l'analyse manuelle de l'environnement, à la recherche de ces signes précoces de changements ou de menaces, devient une opération trop complexe car les veilleurs se retrouvent vite noyés dans la masse d'informations[2]. Dès lors l'utilisation d'un système d'information adapté à l'activité de veille devient nécessaire, tant pour collecter que pour filtrer ces informations mais également pour gérer les différentes connaissances mises en jeu par les veilleurs lors de leur tâche d'analyse. Dans cet article, nous nous intéresserons dans une première partie aux solutions apportées par une plateforme *open source* d'intégration et de construction d'applications de veille stratégique sur des sources ouvertes d'information. Nous proposerons alors des solutions pour permettre l'interopérabilité d'outils variés comme des outils de fouille de documents ou des outils de gestion de connaissances et également des solutions pour proposer aux veilleurs des applications personnalisables et adaptées à leurs besoins. Dans une seconde partie, nous proposerons une approche pour aider les veilleurs dans la détection de signaux faibles. Nous étudierons alors comment la construction assistée de scénarios opérationnels peut être exploitée pour capitaliser des connaissances expertes ou tacites, mises en jeu dans des mécanismes de cognition comme la projection mentale[3]. Nous proposerons un outil de captage et de formalisation de ces connaissances permettant la construction graphique de scénarios ainsi que leur stockage dans une base de connaissance, conformément aux standards du web sémantique[4] proposés par le W3C<sup>1</sup>. Nous verrons ensuite comment gérer et exploiter les scénarios au moyen d'un cycle d'exploitation dédié. Ce dernier permettra de tirer profit de ces connaissances capitalisées dans les différentes grandes étapes du processus de veille[3] comme la collecte, la sélection et la diffusion des nouvelles informations ainsi que dans les phases de gestion de la mémoire d'entreprise et de construction du sens.

---

<sup>1</sup> World Wide Web Consortium

# 1 Introduction

Avec la croissance d'Internet et l'émergence de nouvelles sources d'information ouvertes et accessibles à tous, de nouveaux systèmes d'information plus performants sont désormais nécessaires pour accéder et traiter ces informations disponibles. De plus, dans le cas de la veille stratégique, ces systèmes doivent pouvoir être capables de détecter les informations pouvant constituer des signes précoces de changement ou de menace. Dès lors, le développement et l'utilisation de tels systèmes deviennent des tâches de plus en plus complexe face au nombre croissant de gisements d'information, d'outils et de méthodes disponibles aujourd'hui. Dans cet article, nous identifierons dans un premier temps, les principales problématiques rencontrées lors de la mise en place d'un système de veille capable de traiter des informations issues de sources ouvertes, puis dans un second temps, nous proposerons des solutions en étudiant la mise en place d'un système de veille au moyen d'une plateforme d'intégration adaptée. Enfin, nous proposerons une approche pour la capitalisation de connaissances expertes au moyen de scénarios opérationnels, qui permettra d'améliorer la détection de signaux faibles dans un cycle de veille.

## 2 Problématiques

### 2.1 Problématiques liées aux sources d'information ouvertes

Le développement des moyens de communication, notamment par le biais du Web 2.0, favorise l'émergence de nouvelles sources d'information de plus en plus complexes à traiter dans le cadre d'une activité de veille stratégique. En effet, ces nouvelles sources mettent à disposition des informations possédant des caractéristiques particulières que nous tenterons d'identifier. La prise en compte de ces caractéristiques est primordiale pour obtenir un système adapté, à la fois aux besoins des veilleurs mais également aux informations que le système est amené à traiter. Ainsi, les principales problématiques rencontrées lors du traitement des informations issues de sources ouvertes sont liées aux points suivants.

- La quantité d'information à traiter : la quantité et le volume d'informations mises à disposition aujourd'hui sur des sources ouvertes, notamment avec la croissance des contenus vidéos en ligne, sont tels qu'il devient impossible de collecter manuellement toutes ces informations. En effet, la collecte de ces grands volumes nécessiterait des connexions réseaux rapides, du temps, ainsi qu'un espace de stockage important, dont on ne dispose généralement pas. De plus, face à cette quantité d'informations disponibles, le veilleur se retrouve noyé et il devient impossible pour lui de traiter efficacement ces nouvelles données, et de discerner clairement les informations clés. Comment traiter ces volumes importants de données ?
- L'hétérogénéité des formats et structures d'informations : les informations disponibles sont proposées dans des formats variés (pages HTML, flux RSS<sup>2</sup>, réseaux sociaux, wiki, blogs, forums, etc.) et ne sont pas toujours structurées. Le traitement par les veilleurs de ces ressources passe par l'utilisation d'un ensemble varié d'outils, souvent non interopérables entre eux. Pour utiliser convenablement ces outils le veilleur doit se former à leur utilisation combinée, augmentant encore la complexité de son travail. Comment réduire la complexité du système d'information tout en restant ouvert aux formats et structures de données existants ou en développement ?
- La qualité et l'interprétation des informations : les informations disponibles sur des sources ouvertes peuvent être peu fiables, contradictoires, dispersées, et il est difficile, voire impossible de savoir quel crédit leur accorder au premier abord. Il convient alors de recouper et d'analyser ces informations pour leur donner un sens et une valeur, ce qui reste une étape manuelle et donc couteuse, à la fois en terme de temps et de moyens. Comment rassembler et sélectionner efficacement dans la masse d'informations, les plus pertinentes, qui seront ensuite interprétées et auxquelles on tachera de donner un sens et évaluer une crédibilité ?

---

<sup>2</sup> Really Simple Syndication

## 2.2 Problématiques liées à la conception d'un système de veille adapté

Aujourd'hui, les outils d'aide à la veille tentent de répondre à ces premières problématiques, mais du fait de l'évolution constante des technologies, des formats et des structures d'information, ces outils ne sont pas toujours cohérents et évolutifs. Exploiter ces outils dans un processus de veille pose également de nombreux problèmes, parmi lesquels :

- Le choix des outils et des technologies : l'essor des nouvelles technologies s'accompagne d'une diversité croissante des outils mis à la disposition des veilleurs, que ce soit par les communautés scientifiques ou par les éditeurs spécialisés. L'offre en outils de traitement de l'information est telle qu'il devient difficile pour le veilleur de déterminer quels sont les outils dont il a vraiment besoin. De plus, la prise en main de ces outils peut s'avérer coûteuse en temps pour les veilleurs qui doivent s'approprier l'outil pour en tirer profit. Comment suivre, choisir et profiter des nouvelles technologies et des nouveaux outils sans bouleverser les méthodes de travail des veilleurs ?
- L'interopérabilité des outils : nombre de logiciels disponibles sont souvent mal adaptés au veilleur car ils sont soit trop spécialisés, c'est-à-dire qu'ils offrent des fonctionnalités intéressantes mais difficiles à prendre en main, soit ces logiciels fournissent beaucoup de fonctionnalités mais peu d'entre elles sont réellement intéressantes. Dès lors il serait intéressant de combiner au sein d'une même application, les fonctionnalités intéressantes des différents outils existants. Cependant, ces outils n'offrent que peu ou pas de possibilité d'interfaçage entre eux. En effet, leur développement est réalisé indépendamment les uns des autres et chacun possède son propre modèle de donnée interne. Sans méthodologie dédiée[5], il est souvent impossible d'intégrer directement les fonctionnalités intéressantes des différents outils au sein d'une même application plus cohérente. De même, il est souvent nécessaire d'entreprendre des développements spécifiques et complexes, hors de portée des compétences du veilleur. Comment intégrer facilement, au sein d'une même application, les fonctionnalités intéressantes d'un ensemble d'outils hétérogènes et non interopérables ?
- L'ergonomie et l'interface utilisateur : en plus de disposer de fonctionnalités issues de différents outils, il est nécessaire que le veilleur dispose d'un système clair, cohérent et ergonomique. En effet, pour appréhender la complexité du système, le veilleur doit comprendre son fonctionnement et accéder facilement aux fonctionnalités dont il a besoin. De même, de nombreux utilisateurs demandent maintenant que l'interface graphique de l'application qu'ils utilisent soit personnalisable. Comment intégrer de façon ergonomique et personnalisable, au sein d'une même application, des IHM<sup>3</sup> provenant d'outils variés ?

## 2.3 Problématiques liées à la gestion de connaissances expertes

Le processus de veille exploite les nombreuses connaissances du veilleur ou de l'expert, que ce soit par le biais de son expérience, de sa connaissance des domaines et des thèmes impliqués dans l'activité de veille, ou des connaissances mises en jeu lors du processus de création du sens[6]. De même, sa capacité à raisonner et à synthétiser l'information implique l'usage de processus cognitifs reposant eux même sur des connaissances variées. Par conséquent, pour mettre en place un système d'aide à la veille adapté et performant, il convient de prévoir la mise en place d'outils dédiés à la gestion de l'ensemble de ces connaissances au sein de ce système. Les problématiques liées à la gestion de la mémoire de l'entreprise[7] sont nombreuses et au cœur de nombreuses recherches. Nous résumons ici les principales problématiques, particulièrement impliquées dans le cas de la veille stratégique auxquelles nous tenterons de fournir une solution par la suite :

- La construction des connaissances : les experts utilisent des connaissances tacites qu'ils ont souvent du mal à exprimer ou à retranscrire au sein d'un système de gestion des connaissances. De même, la construction de façon collaborative d'une connaissance d'entreprise, partagée et acceptée de tous, reste une tâche longue et complexe du fait, par exemple, des nombreux points de vue, parfois divergents, exprimés par les experts. Comment faciliter le captage des connaissances expertes au sein du système de gestion de connaissance ?

---

<sup>3</sup> Interface Homme Machine

- La mise à jour des connaissances : par définition, l'activité de veille s'intéresse aux évolutions d'un environnement changeant, où l'on observe de nouvelles situations et où l'on exploite un ensemble d'informations et de connaissances en rapport avec ces changements. Par conséquent, les connaissances impliquées dans le processus de veille évoluent elles-mêmes en permanence. Un système d'aide à la veille efficace doit donc permettre la mise à jour de ces connaissances. De la même façon que pour leur construction, la mise à jour de façon collaborative des connaissances peut constituer une tâche longue et complexe. Face aux changements rapides de l'environnement, comment mettre à jour efficacement ces connaissances, de façon collaborative ?
- L'exploitation des connaissances : pour impliquer l'expert ou le veilleur dans la capitalisation des connaissances, il est nécessaire que le système exploite ces connaissances et présente à l'utilisateur un résultat pertinent issu de cette exploitation, justifiant alors l'effort de capitalisation. Il convient donc de formaliser ces connaissances de façon à les rendre compréhensibles et exploitables par le système. Comment formaliser des connaissances expertes et les rendre exploitables par le système et ainsi justifier l'effort de capitalisation ?

### 3 Plateforme d'intégration WebLab©

La plateforme d'intégration *open source* [WebLab](#), développée par EADS, vise à faciliter l'intégration de composants logiciels et le développement d'applications exploitant des outils et des techniques de fouille de documents multimédia[8]. C'est au travers de nombreux projets de recherche comme [WebContent](#) ou VITALAS<sup>4</sup> et des projets opérationnels que le département IPCC<sup>5</sup> développe depuis 5 ans cette plateforme et son expertise sur le traitement des informations non structurées et sur le web sémantique. Le traitement journalier de plusieurs milliers de pages Web dans les applications mises en place ou l'indexation et l'accès à plusieurs millions d'images, effectués dans le cadre du projet VITALAS, ont notamment validé les choix techniques retenus pour le développement de la plateforme. Au sein de ces projets, les prototypes ou les applications développés s'appuient tous sur la plateforme WebLab qui offre alors l'architecture et des solutions techniques variées comme par exemple, le format pivot pour l'échange et la représentation de document multimédia (basé sur XML<sup>6</sup> et détaillé par la suite), le portail Web basé sur l'outil [Liferay](#) pour l'IHM, ou encore l'ESB<sup>7</sup> ([Petals](#)) qui orchestre et fait collaborer les différents services développés par les partenaires de ces projets (laboratoires, organismes, etc.). Nous étudierons par la suite comment ces choix techniques facilitent l'intégration et la combinaison de composants logiciels hétérogènes (COTS<sup>8</sup> ou open source), rendant dès lors accessibles aux utilisateurs de nombreuses fonctionnalités comme la collecte de site web, la reconnaissance d'entités nommées, la traduction, l'indexation de documents, l'analyse sémantique de données, etc. Dans la suite de cet article, nous présenterons les solutions plus globales qu'apporte la plateforme WebLab pour les différentes problématiques liées à la veille et présentées précédemment.

#### 3.1 Solutions techniques de la plateforme WebLab

La plateforme apporte des solutions techniques par rapport aux problématiques citées précédemment de par sa décomposition en trois couches, illustrées par la figure 1 et expliquées ci-dessous :

- Les WebLab Applications sont le résultat de l'intégration des différents composants sous la forme d'une application. La construction des applications WebLab est facilitée et personnalisable grâce à l'utilisation d'un portail Web permettant la composition dynamique de fragments d'IHM (au moyen de la

<sup>4</sup> Video & image Indexing and reTrievAl in the LARge Scale, <http://vitalas.ercim.org/>

<sup>5</sup> Information Processing Control and Cognition

<sup>6</sup> Extensible Markup Language

<sup>7</sup> Enterprise Service Bus

<sup>8</sup> Commercial off-the-shelf (composant pris sur étagère)

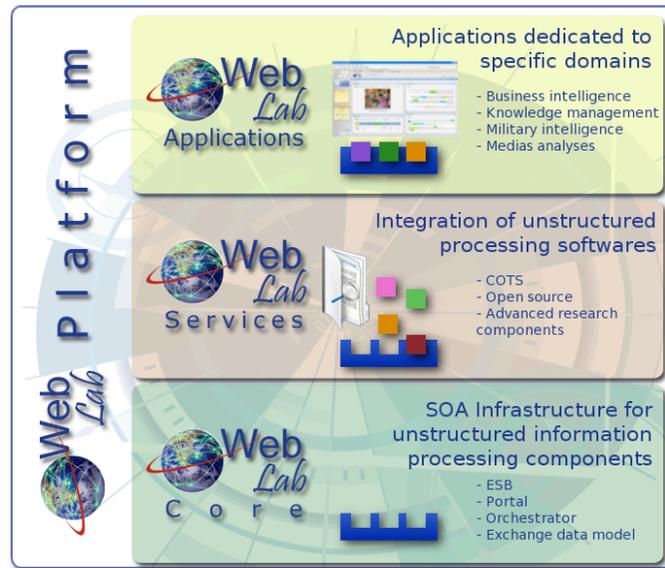


Figure 1: Les couches de la plateforme WebLab

technologie Portlet), apportant ainsi une solution aux problématiques liées à l'ergonomie, à la personnalisation des applications et à l'intégration d'IHM en provenance d'outils variés. Il est alors possible d'accéder soit à l'IHM native de l'outil (portlet « proxy ») soit d'exploiter les différents services au moyen de portlets spécialisés. Les composants et fonctionnalités intégrés au portail sont donc utilisables conjointement, au sein d'une seule et même application cohérente.

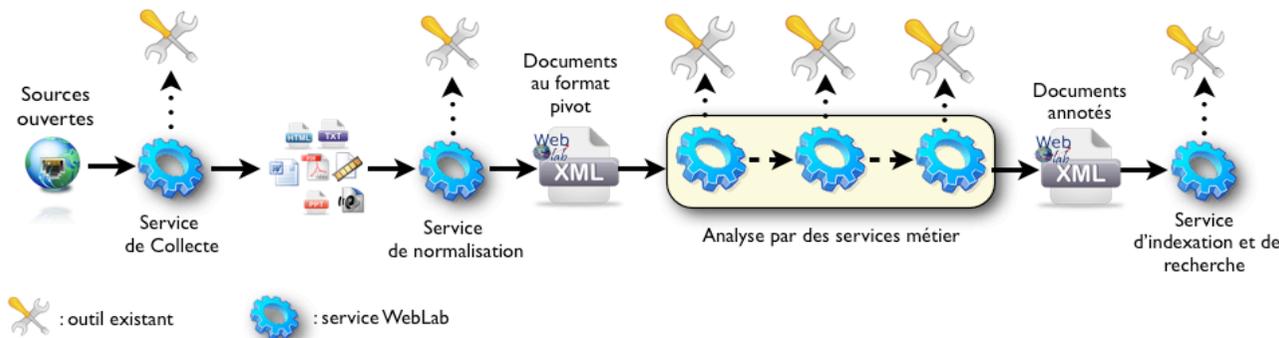
- Les WebLab Services qui constituent un ensemble cohérent de briques logicielles, exposées sous la forme de services Web (accessibles par le protocole SOAP<sup>9</sup>), directement intégrables à l'application grâce à l'utilisation d'interfaces normalisées, spécialisées pour le traitement de documents multimédias et décrites en WSDL<sup>10</sup>. Ces services WebLab donnent accès de façon indirecte aux différentes fonctionnalités proposées par les COTS, les composants *open source*, ou les composants développés par EADS et ses partenaires. Par exemple, un composant de détection de la langue dans le texte comme [Ngramj](#) ou un composant d'extraction d'entité nommée dans le texte comme [GATE](#)[9], se verra intégré dans un service implémentant l'interface « Analyser ». L'utilisation de ces interfaces rend possible l'ajout dans l'application, à tout moment, d'un nouveau service. Elles donnent également la possibilité d'intervenir directement un composant par un autre remplissant la même fonction (cas par exemple du passage d'un composant *open source* à un composant commercial plus performant).
- Le WebLab Core est la base technique *open source* qui fournit l'environnement nécessaire au développement, à l'intégration et à l'exploitation des différents services et portlets WebLab. Les choix techniques retenus ici ont pour objectif de permettre et de faciliter l'interopérabilité technique entre les composants

<sup>9</sup> Simple Object Access Protocol

<sup>10</sup> Web Services Language Description

WebLab. Pour ce faire la plateforme propose une architecture SOA<sup>11</sup> et surtout, un modèle d'échange de données, qui permet de représenter, sous une forme XML, tout type de document (texte, audio, vidéo, image, etc.) issu d'une source ouverte. Le choix du format XML apporte ici la flexibilité nécessaire à la représentation de tout type de document et suit les recommandations W3C. Ce format d'échange est une solution aux problématiques liées à l'hétérogénéité des formats et des structures de données. En effet, les services de l'application traitent et s'échangent des documents normalisés dans un seul et même format pivot, en faisant abstraction de leurs formats d'origine. La transformation des données accessibles en sources ouvertes vers des documents au format pivot est alors effectuée au moyen de services de normalisation, basés sur des outils existants comme par exemple [Tika](#). L'ensemble des connaissances extraites sur ces documents normalisés par les outils (titre, auteur, contenu, langue, entités extraites, etc.) sont décrites sémantiquement, conformément à des ontologies de domaine (comme par exemple l'ontologie [DublinCore](#) pour les métadonnées) et directement intégrées au document pivot au moyen d'annotations au format RDF<sup>12</sup>/XML. Les documents produits peuvent alors être enrichis sémantiquement au fur et à mesure de leur traitement par les différents services métiers. Cette formalisation des connaissances dans un langage du web sémantique au sein des documents permet d'exploiter de nombreuses fonctionnalités fournies par les outils de gestion de connaissances comme l'exploitation d'ontologies de domaine, l'interrogation des bases de données sémantiques, etc. De fait, la plateforme est tournée vers le web sémantique où les données sont interprétables sémantiquement par des machines, améliorant leur traitement. Une description plus détaillée du format d'échange[8] ainsi que des exemples de documents pivots montrent que la plupart des informations disponibles sur le Web, ainsi que les connaissances extraites sur ces derniers sont modélisables dans ce format pivot.

Pour automatiser la collecte et le traitement des ressources Web, la plateforme WebLab met à disposition des chaînes de traitement adaptées, exprimées en BPEL<sup>13</sup> et exécutées par le moteur d'orchestration de l'ESB (qui se charge d'appeler chaque service, de véhiculer les documents de service en service, de répartir la charge entre les services, etc.). La figure 2 illustre le déroulement d'une chaîne de traitement générique au sein d'une application WebLab et détaillée comme suit : (1) un service de collecte (basé par exemple sur l'outil [HTTrack](#)) est appelé pour collecter des ressources du Web dans différents formats (pages HTML, PDF, fichiers vidéo, etc.); (2) un service de normalisation est appelé pour transformer chacune de ces ressources collectées en un document au format pivot; (3) chaque document normalisé est ensuite traité et enrichi sémantiquement (annotation) par un ensemble de services d'analyse (détection de la langue, traduction, résumé, etc.); (4) enfin, chaque document traité est transmis à un service d'indexation (basé par exemple sur l'outil [Lucene](#)). Les documents traités sont alors accessibles au sein du portail Web au moyen de portlets spécialisés (portlet de recherche et portlet de visualisation de document).



<sup>11</sup> Service Oriented Architecture

<sup>12</sup> Resource Description Framework

## 3.2 Apport de la plateforme dans une application de veille stratégique

Dans le cas d'une application de veille, la plateforme apporte en particulier des solutions pour les problématiques suivantes :

- Le choix des outils et des technologies : les fonctionnalités implémentées par les différents outils sont exposées sous la forme de services web, ce qui permet d'abstraire pour le veilleur, aussi bien les modèles de données internes des outils, que les technologies et formats qu'ils exploitent. L'ensemble des fonctionnalités de ces outils est disponible au sein d'une seule application, accessible depuis un navigateur Web, ce qui facilite la prise en main de ces outils par le veilleur. Dès lors, l'ajout d'un nouvel outil ou l'exploitation d'une nouvelle technologie nécessite simplement le développement d'un adaptateur exposant ses fonctionnalités sous la forme d'un service web, conformément à l'une des interfaces du WebLab Core. De nombreux adaptateurs sur des outils existants du monde *open source* ou commerciaux sont déjà disponibles avec la plateforme WebLab, minimisant cet effort de développement. Le veilleur dispose alors d'une application flexible et personnalisable lui permettant de suivre l'évolution des technologies Web et d'ajouter de façon incrémentale des fonctionnalités et de nouveaux outils sans bouleverser ses habitudes de travail.
- L'interopérabilité des outils : la mise en place de chaînes de traitement de l'information, spécifiques et adaptées, permet la combinaison des différents services exploités au sein des processus de traitement, ce qui assure donc l'interopérabilité des outils. Le veilleur peut alors combiner automatiquement les fonctionnalités des différents outils grâce aux différentes chaînes de traitement proposées et traiter en masse les documents impliqués dans le processus de veille par l'ensemble des outils mis à sa disposition.
- Le dimensionnement de l'application face aux volumes de données à traiter : des services de collecte d'information permettent la collecte automatique des informations disponibles en sources ouvertes, automatisant la tâche de collecte. De plus, la flexibilité apportée par l'architecture SOA, qui autorise l'ajout en temps réel de nouveaux services, permet de redimensionner en direct l'application et de répartir géographiquement les besoins réseau et les volumes de stockage (multiplication et redondance des services). Le veilleur peut donc facilement adapter le périmètre de la veille et l'application face aux changements de l'environnement pour rapidement disposer d'informations nouvelles.

## 4 Aide à la détection de signaux faibles par modélisation de scénarios

Pour répondre aux problématiques liées à la gestion de connaissances expertes exprimées précédemment, et pour aider les veilleurs au cours des différentes étapes du processus de veille, nous proposons une approche basée sur la construction et l'évaluation dynamique de scénarios opérationnels. Cette modélisation permet, par exemple, d'améliorer la reconnaissance de situations<sup>[10]</sup> et donc d'aider les veilleurs dans la détection de signaux faibles. Nous définissons ici un signal faible comme un signe précoce de changement ou de menace. La détection de signaux faibles est l'un des enjeux majeur d'un processus de veille stratégique car c'est la détection au plus tôt de ces signes qui permettra au décideur de prendre des décisions face aux changements ou aux menaces survenant dans l'environnement.

Nous définissons un scénario comme une modélisation experte à propos d'un ensemble de situations pouvant survenir dans l'environnement, ainsi que les différents liens de cause à effet, existants entre ces situations. Dans l'arbre de gauche de la figure 3, l'exemple d'un scénario opérationnel où est modélisé un ensemble de situations (rectangles violets) susceptibles de se réaliser dans l'environnement, ainsi que les liens de cause à effet entre ces situations (flèches vertes). Dans un scénario, à chacune des situations modélisée, nous définissons un état de réalisation (la situation est réalisée, non réalisée, ou sa réalisation est inconnue). La réalisation d'une situation peut alors être fixée manuellement par l'expert ou évaluée automatiquement par le système. Dans ce cas, une situation est évaluée comme réalisée par le système si ses situations « causes » (par exemple, les situations impliquées dans une relation de cause à effet avec cette situation) sont elles-mêmes réalisées. Afin de rendre possible la reconnaissance des différentes situations, nous définissons, pour chaque situation, un ensemble de contraintes. Ces contraintes

---

<sup>13</sup> Business Process Execution Language

sont ici utilisées par l'expert pour décrire d'une part les composantes de la situation, c'est-à-dire ses acteurs, la zone géographique où elle se réalise, etc. (rectangles verts de la figure 3) et d'autre part pour fixer d'éventuelles conditions à l'avènement de la situation comme l'évaluation d'indicateurs ou de tendances. L'arbre droit de la figure 3 illustre une modélisation experte de ces contraintes définissant une situation. Dans l'exemple, la situation est définie par deux contraintes typées (rectangles bleus) reliées entre elles par une condition (ici une combinaison en orange), impliquant des concepts (rectangles verts). L'analyse de ces contraintes sera utile pour déterminer l'état de réalisation de la situation de façon automatique par le système. En ayant une vision claire de l'état de réalisation des situations, l'expert peut facilement suivre leur évolution dans le temps. Lors de la définition des contraintes, l'expert exploitera au maximum des concepts (rectangles verts de la figure 3) issus de bases de connaissances adaptées aux domaines impliqués dans la veille, pour en fournir une description non ambiguë. Nous détaillerons dans la suite de cet article, les différentes bases de connaissances existantes, exploitées aujourd'hui pour la construction des scénarios. Enfin, pour s'adapter rapidement aux changements de l'environnement, un scénario est modifiable à tout instant par l'expert pour y ajouter ou y supprimer des situations, des liens de cause à effet ou des contraintes.

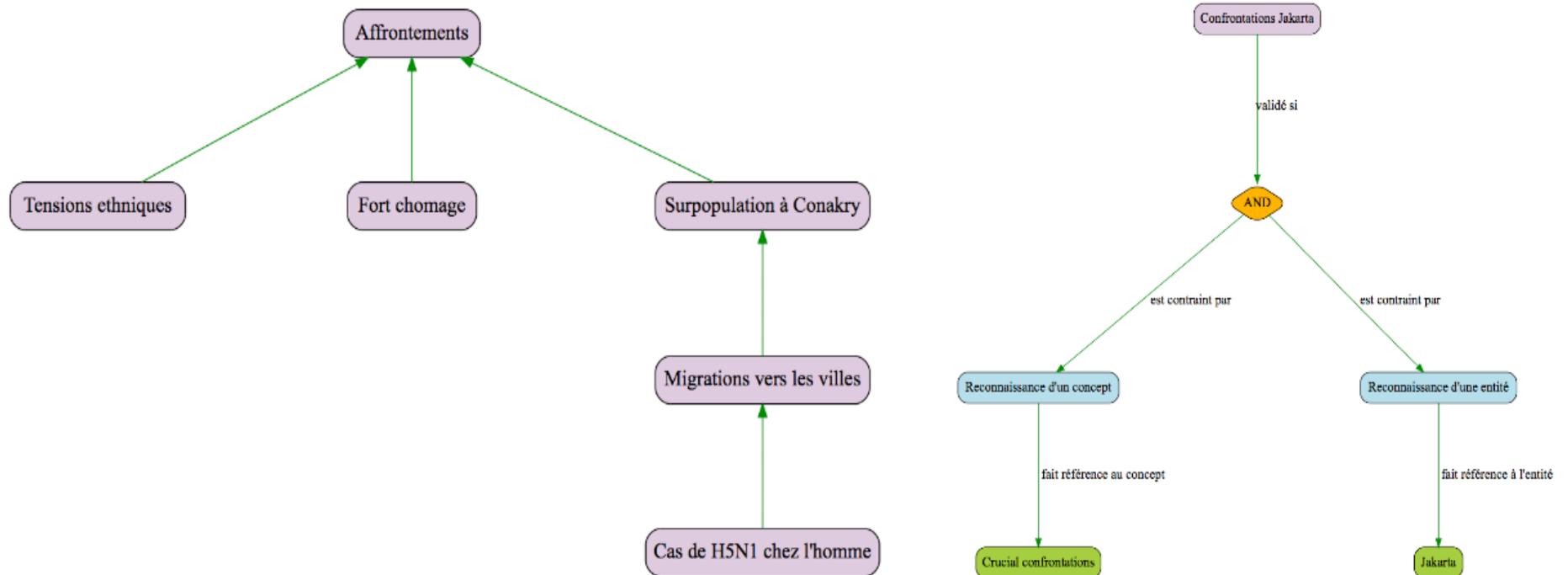


Figure 3: Exemple d'un scénario opérationnel simple et d'un arbre de contraintes sur une situation

## 4.1 Capitalisation de connaissances par construction de scénarios

Ainsi défini, nous pensons qu'un scénario opérationnel permet de capitaliser de nombreuses connaissances :

- Des connaissances expertes à propos de situations passées : dans le cas où l'expert construit des scénarios en se fondant sur son expérience dans le but de détecter si des situations déjà rencontrées dans le passé se reproduisent dans l'environnement. Ici l'expert regarde plutôt dans le passé et exprime, au travers du scénario, des connaissances sur des événements antérieurs et sur son analyse *a posteriori* des événements.
- Des connaissances expertes à propos de situations futures : dans le cas où l'expert construit des scénarios en se fondant plutôt sur ses intuitions ou des indices qu'il possède dans le but de détecter si de potentielles situations attendues se produisent. Dans ce cas, l'expert exploite le mécanisme cognitif de projection mentale et exprime au travers du scénario, des connaissances à propos de possibles situations futures et des connaissances tacites sur son intuition personnelle ou sur son analyse du contexte environnemental.

Pour faciliter la construction des scénarios, nous proposons un ensemble de composants WebLab (portlet et services), permettant de créer et de visualiser graphiquement des scénarios sous forme de graphes, orientés et acycliques. Le portlet fournit l'IHM pour construire et visualiser les scénarios et les contraintes. Dans cet outil de construction, nous représentons les scénarios et les contraintes comme expliqué précédemment et illustré dans la figure 3. Les services permettent, de stocker les scénarios dans une ontologie (décrite ci-dessous) et d'exploiter ces connaissances pour l'analyse de données issues de sources ouvertes.

## 4.2 Ontologie des scénarios

Les scénarios modélisés au moyen de l'outil de construction sont directement ajoutés dans une ontologie dédiée à la description de scénario. Cette ontologie est elle-même gérée dans une base de connaissance au moyen de l'outil [Sesame](#) qui assure le stockage potentiel de milliers de scénarios. L'ontologie proposée est exprimée au format OWL<sup>14</sup>, conformément aux recommandations du W3C. L'ontologie est composée aujourd'hui de 32 classes nécessaires à la description des scénarios (scénario, situation, types de contraintes, types de concepts, etc.) et 46 propriétés sur ces objets (propriété de cause à effet entre situations, propriété entre une situation et une contrainte, etc.). L'ontologie se décompose en deux niveaux :

- Le premier niveau, stable, qui décrit l'ensemble minimal des classes et des propriétés, nécessaire à la représentation de scénarios génériques. Nous définissons ici un scénario générique comme un scénario n'impliquant aucune contrainte typée ni aucun concept de domaine métier. Sont par exemple définis à ce niveau de l'ontologie, les classes *Scenario* (classe des scénarios), *State* (classe des situations), *Constraint* (classe générique des contraintes), *SimpleConstraint* (classe générique des contraintes simples), *CompositeConstraint* (classe générique des contraintes composites), *ConstraintSatisfaction* (classe générique des preuves de la réalisation des contraintes), ainsi que les propriétés entre ces classes comme par exemple la propriété *isConstraintBy* liant une situation à la contrainte la décrivant.
- Le second niveau de l'ontologie, extensible, étend le premier niveau et apporte les définitions complémentaires qui permettent de typer les contraintes et de définir les objets en lien avec ces contraintes. Sont par exemple définis à ce niveau, les sous-classes de contraintes génériques comme la contrainte de type *EventRecognition* (reconnaissance d'un événement), la classe *Event* (événement) associée à ce type de contrainte, ainsi que les propriétés et les objets du domaine liés à un événement (comme l'acteur, la cible, le type de l'événement, etc.).

La décomposition en deux niveaux offre ici la possibilité d'ajouter de nouveaux types de contraintes comme par exemple la reconnaissance ou l'évaluation d'indicateurs, sans modification dans l'outil de construction de scénario qui exploite, lui, le premier niveau stable de l'ontologie. La création de scénarios au travers de l'outil de construction peuple l'ontologie en produisant des instances de ces classes et propriétés. La construction et le peuplement d'une ontologie de domaine

---

<sup>14</sup> Ontology Web Language

étant une étape complexe et coûteuse en temps, nous proposons d'exploiter les ontologies de domaine déjà existantes et disponibles aujourd'hui sur le Web, notamment par le biais de projets comme [LinkedData](#). Ainsi, les différentes composantes des contraintes typées et exprimées dans les scénarios (acteurs, lieux, organisation, etc.), référencent directement des concepts décrits dans l'ontologie [DBpedia](#) (plus de 2 millions de concepts) ou dans l'ontologie [GeoNames](#) (environ 6,5 millions de concepts géographiques). En ce sens, l'ontologie dédiée à la modélisation de scénario que nous proposons permet de profiter des connaissances existantes du Web sémantique. La flexibilité apportée par l'utilisation des ontologies permet également d'envisager l'ajout et la modélisation de nouveaux types de contraintes, d'indicateurs ou de concepts au sein des scénarios.

### 4.3 Cycle d'exploitation des scénarios

Les possibilités d'exploitation des scénarios dans les différentes étapes du processus de veille sont nombreuses et extensibles. En effet, du fait de la formalisation sémantique des scénarios et de la flexibilité apportée par la plateforme WebLab, il est possible d'ajouter de nombreux services WebLab chargés d'analyser la base de connaissance des scénarios pour aider les veilleurs dans les étapes importantes de la veille. Pour exploiter au mieux les scénarios, nous proposons un cycle en 6 étapes allant de leur construction et leur mise à jour à leur exploitation dans un processus de veille. La figure 4 illustre ce cycle et nous détaillons ci-dessous les différentes étapes de ce cycle et ses apports potentiels dans les phases du processus de veille, ainsi que l'inventaire des outils aujourd'hui existants pour assister le veilleur :

- Etape 1 : construire et formaliser des scénarios. Cette étape permet d'améliorer la gestion de connaissances et la mémoire de l'entreprise et stimule l'imagination de l'expert. Cette étape est aujourd'hui rendue possible par l'utilisation d'un portlet de construction de scénario.
- Etape 2 : utiliser les scénarios pour focaliser la collecte. On améliore ici la phase de ciblage et de collecte de l'information en exploitant par exemple l'état de réalisation des situations, pour récupérer entre autre des compléments d'informations sur les situations *a priori* imminentes ou pour explorer de nouvelles thématiques. A cette étape, il est possible de refocaliser la collecte en consultant l'état de réalisation des situations dans le portlet de visualisation des scénarios. Les services de collecte existants permettent l'automatisation de la collecte mais leur configuration (gestion des sources de collecte) reste manuelle.
- Etape 3 : utiliser les outils disponibles dans la plateforme WebLab pour analyser les nouveaux documents et en fournir une description sémantique. L'utilisation des connaissances modélisées au sein des scénarios par des outils de reconnaissance d'entité permet de focaliser l'extraction des concepts dans les documents collectés, de façon automatique.
- Etape 4 : utiliser les descriptions sémantiques des documents pour vérifier si des contraintes des scénarios sont satisfaites. On améliore ici la phase de sélection des documents en mettant à profit les connaissances expertes capitalisées et le résultat des outils d'analyse. Un service d'analyse de contraintes permet de faire le lien entre les instances reconnues dans les documents et celles modélisées dans les scénarios de façon automatique (les connaissances produites sont mémorisées dans la base de connaissances).
- Etape 5 : classer les documents et envoyer des alertes. Les documents satisfaisants des contraintes sont jugés comme pertinents pour certaines situations des scénarios et sont signalés aux veilleurs qui pourront valider la pertinence de ces documents. Nous améliorons ici la diffusion des informations. Cette étape est ici automatisée par un service de recherche sémantique de document en liens avec les situations des scénarios et intégré au portlet de visualisation de scénarios.
- Etape 6 : analyser manuellement les documents. Les veilleurs procèdent à l'analyse des documents suggérés puis s'appliquent à donner un sens à ces nouvelles informations. Si l'information permet de déduire que les situations ou les scénarios ont évolué, le veilleur met à jour les scénarios et valide l'état de réalisation des situations. Lors de la création du sens, l'analyste peut s'aider des scénarios pour mieux comprendre l'évolution de l'environnement. Nous améliorons ici la phase de construction du sens. La consultation dans le portlet de visualisation de document affiche les entités reconnues, facilitant l'analyse du document.

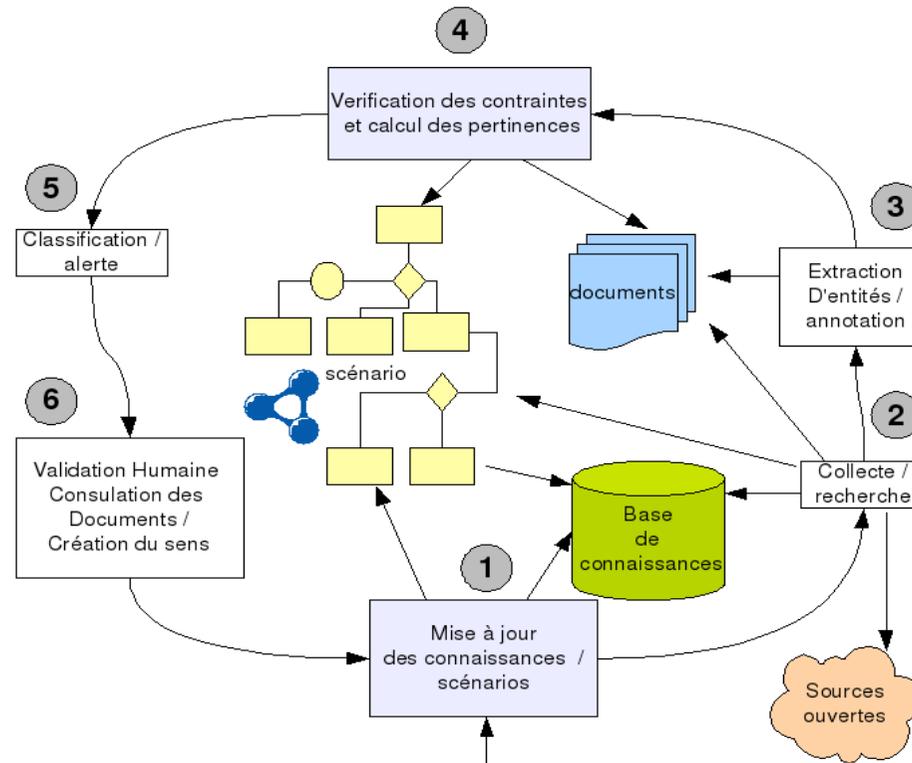


Figure 4: Illustration du cycle d'exploitation des scénarios

## 5 Conclusion et perspectives

Au cours de cet article, nous avons présenté une plateforme d'intégration *open source*, adaptée à la construction d'une application de veille stratégique. Nous avons montré que cette plateforme apporte des réponses à des problématiques que l'on rencontre lors d'une activité de veille sur des sources d'informations ouvertes. Nous avons également souligné la flexibilité de cette plateforme par rapport aux nombreux formats ou structures de données existants. De même, nous avons montré que cette plateforme est particulièrement adaptée à l'intégration des outils existants de fouille de données. Enfin, nous avons exposé une approche pour l'aide à la détection de signaux faibles au moyen de scénarios opérationnels et présenté l'ontologie et les outils basés sur la plateforme WebLab, permettant de les construire, de les formaliser dans une base de connaissance conformément aux standards du web sémantique. Finalement nous avons proposé un cycle qui permet d'exploiter ces scénarios au sein des différentes phases d'un processus de veille ainsi que les outils développés ou existants dans la plateforme WebLab pour automatiser ces étapes. Dans la suite de nos travaux nous étudierons plus en détail la conception de nouveaux services exploitant au mieux la base de connaissance constituée par les scénarios, et notamment des services d'évaluation d'indicateurs par l'analyse de sources ouvertes de connaissances comme par exemple l'ontologie [CIA - Factbook](#). Il serait également pertinent d'étudier les techniques qui permettraient de générer ou de suggérer automatiquement des scénarios opérationnels.

## 6 Bibliographie

- [1] ANSOFF Igor H., *Managing Surprise and Discontinuity - Strategic Response to Weak Signals*, California Management Review, 1975
- [2] LESCA H., SCHULER M., *Comment ne pas être noyé sous les informations*. Économies et Sociétés, Série Sciences de Gestion, n°2/1998, p.159-177
- [3] LESCA H. *Veille stratégique: la méthode LE SCAnning*, EMS, 2003
- [4] BERNERS-LEE T., HENDLER J., *Scientific publishing on the semantic web*, Nature, 2001
- [5] DOUCY J., ABDURLAB H., GIROUX P., KOTOWICZ J.P., *Méthodologie pour l'orchestration sémantique de services dans le domaine de la fouille de documents multimédia*, MajecSTIC, 2008
- [6] LESCA N., *Construction du sens*, Thèse doctorale, 2001
- [7] DIENG, R. et al, *Knowledge management: Méthodes et outils pour la gestion des connaissances*, Dunod 01 informatique, 2005
- [8] GIROUX P. et al, *WebLab: An integration infrastructure to ease the development of multimedia processing applications*, ICSSEA, 2008
- [9] CUNNINGHAM H., *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*, Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002
- [10] DELAVALLADE T., MOUILLET L., BOUCHON-MEUNER B., COLLAIN E, *Monitoring Event Flows and Modelling Scenarios for Crisis Prediction: Application to Ethnic Conflicts Forecasting*, International Journal Of Uncertainty Fuzziness And Knowledge Based Systems, 2007