SEMANTICA ALERTELOR ÎN SISTEMELE DE AVERTIZARE PENTRU SITUAȚII DE URGENȚĂ. PREMISELE UNEI ONTOLOGII A ALERTĂRII ÎN SITUAȚII DE URGENȚĂ

Nicolae-Dorel Constantinescu

Doctorand, Facultatea de Automatică şi Calculatoare,

Universitatea „Politehnica" Bucureşti

dorel_nic@yahoo.com

Rezumat: În acest articol se propune definirea semantică a activității de avertizare în situații de urgență, completând o ontologie (BFO) şi utilizând un limbaj de reprezentare (OWL) consacrate. Adițional se arată cum se poate realiza translatarea ontologiei rezultate în structurile de date ale unui limbaj de programare, de asemenea de largă utilizare (JAVA), în scopul implementării în aplicațiile software pentru avertizare în situații de urgență.

Abstract: In this paper we propose to semantically define the emergency alerting activity, expanding an acknowledged ontology (BFO) and using a well-known representation language (OWL). In addition, we show how to translate the resulted ontology into the data structures of a programming language, also widely used (JAVA), in view of its implementation by the emergency alerting software.

1. Introducere

Situații de urgență se produc în permanență (inundații, alunecări de teren, atentate teroriste etc.) şi au nevoie de un management corespunzător. Avertizarea populației este o activitate ce se manifestă la intersecția etapelor de pregătire şi răspuns, din managementul situațiilor de urgență. Sistemele de avertizare pentru situații de urgență au scopul de a înlesni activitatea de avertizare. În mod clasic, sistemele de avertizare pentru situații de urgență se organizează pe trei niveluri: monitorizare, interpretare, alertare [1]. Suplimentar, am identificat drept necesară introducerea celui de-al patrulea nivel, nivelul de corelare. În sprijinul acestui nivel şi al întregului proces de alertare, s-a impus sporirea gradului de rigurozitate cerut în definirea evenimentelor (ideal precursorilor) şi intrinsec alertelor ce sunt generate pe baza evenimentelor. Întrebările la care ne-am propus să găsim un răspuns precis sunt *unde, când* şi *ce* se alertează. Am avut în vedere în lucrări anterioare dimensiunea spațială [2] şi temporală [3], iar în acest articol ne concentrăm asupra aspectului semantic al problemei alertării în situații de urgență. Ca şi în celelalte cazuri, instrumente din sfera tehnologiei informației par să fie croite special pentru a oferi răspunsuri adecvate la problema în discuție. Astfel, fiind stabilită necesitatea unei bune definiri a alertelor şi faptul că acestea sunt create şi procesate prin intermediul sistemelor informatice, vom încerca să creăm premisele unei ontologii pentru acestea şi în acelaşi timp să stabilim posibilitățile în care ontologia ce va fi creată poate fi tradusă în structurile de date ale limbajelor de programare, în scopul oferirii posibilității de manipulare programatică. Vom vedea că dinamica spațio-temporală a alertelor creează particularități de reprezentare a acestora.

2. Aspecte terminologice

Deşi conceptul de *ontologie* este destul de frecvent utilizat, sensul acestuia nu este de fiecare dată foarte bine înțeles. Se ştie că termenul a fost împrumutat din filozofie, unde, în termeni concişi, reprezintă o teorie generală a existenței. Din punctul de vedere al cercetătorului în domeniul Inteligenței Artificiale, o ontologie este o specificație explicită a unei conceptualizări [4]. În teoria Sistemelor Bazate pe Cunoştințe, o ontologie este un catalog al tipurilor de lucruri ce se admite că există într-un anume domeniu de interes, din perspectiva persoanei care utilizează un anume limbaj pentru a vorbi despre acel domeniu [5].

Din punctul nostru de vedere, sarcina de a oferi alerte de urgență implică interoperabilitatea, adică compatibilitatea sau colaborarea naturală a mai multor entități. Pentru aceasta, la nivel semantic, este necesară o abordare comună a alertelor. Tema ontologiei de care avem nevoie nu

este „existența” în general, ci lucruri specifice şi concrete, cum sunt evenimentele, riscurile, situațiile cu caracter de urgență, relațiile dintre acestea etc. Astfel vom înțelege ontologia alertelor ca o descriere explicită şi cât mai clară a conceptelor ce le caracterizează şi a relațiilor ce le leagă. Dezvoltarea unei ontologii pentru un domeniu precum alertarea în situații de urgență este o sarcină ambițioasă pentru a cărei realizare completă este necesar efortul concertat al tuturor actorilor implicați în managementul situațiilor de urgență. În această lucrare încercăm sa facem primii paşi în crearea unei ontologii a alertării în situații de urgență, să identificăm principalele tipuri evenimente, să oferim o modalitate de codificare şi să exemplificăm modalitatea în care aceasta poate fi translatată într-o formă accesibilă programatic. Există inițiative de a crea ontologii în domeniul alertării în situații de urgență (Common Alerting Protocol [6], EDXL DE [7], Tactical Situational Object [8], Philippe Kruchten et al. [9], Paola Di Maio [10], Alessio Malizia et al. [11], Wei Xu et al. [12], Paul Ngo et al. [13]), dar nu am găsit niciuna completă, având în prim plan alertele folosite în sarcina de alertare a populației, cu atât mai puțin una adecvată specificului situațiilor de urgență din România.

3. Ontologiile şi dezvoltarea acestora

De domeniul dezvoltării ontologiilor se ocupă aşa numita inginerie ontologică. Pentru a crea o ontologie, mai întâi sunt colectate toate cerințele privind ontologia respectivă. În mod tipic pentru ingineria ontologică, inginerii şi experții din domeniul respectiv se reunesc într-o echipă ce lucrează conjugat pentru a descrie domeniul şi scopul ontologiei, direcțiile de proiectare, sursele de cunoştințe disponibile, utilizatorii potențiali, cazurile de utilizare şi aplicațiile suportate de ontologie. Rezultatul acestei faze este o descriere cvasi-formală a ontologiei. În a doua fază, cea de rafinare, echipa extinde prin câteva iterații descrierea cvasi-formală şi o formalizează într-un limbaj de reprezentare potrivit. Rezultatul acestei faze este o ontologie matură (cunoscută şi sub numele de ontologia țintă). În a treia fază, ontologia țintă trebuie să fie evaluată conform specificațiilor despre cerințe. În mod curent această fază serveşte ca o dovadă a utilității ontologiei proiectate şi poate implica atât echipa inginerească cât şi utilizatorii finali ai aplicației țintă. Rezultatul acestei faze este o ontologie evaluată, pregătită pentru utilizarea într-un mediu de producție.

Pentru a descrie semantica unui domeniu, sensul datelor despre acesta, sunt diferențiate două modalități de explorare: de sus în jos şi de jos în sus. Dezvoltarea de sus în jos urmăreşte ghidajul teoretic pentru exprimarea semanticii *explicite* a datelor. Domeniul de interes este inițial examinat la un nivel înalt, abstract, apoi modelul astfel identificat este aplicat pentru a descrie conceptele de la nivelurile inferioare [14]. În contrast, dezvoltarea de jos în sus exploatează semantica *implicită* a datelor, bazându-se de exemplu pe specificarea sursei datelor şi a contextului de lucru pentru a explica înțelesul datelor. O metodologie de dezvoltare trebuie să exploateze la nivel înalt conceptele cumva filozofice aplicabile domeniului, cuplate cu elemente materiale, empirice, concrete, specifice domeniului pentru construcția de nivel scăzut şi implementarea ontologiei [15]. Ontologiile formale tratează categorii şi relații care sunt prezente în toate domeniile şi care sunt în principiu aplicabile din orice perspectivă a realității. O astfel de ontologie este BFO (Basic Formal Ontology), o teorie a structurilor de bază ale realității. Modelarea timpului este o preocupare importantă ce generează o dihotomie în privința perspectivelor reprezentate de BFO. BFO identifică mai întâi o perspectivă conform căreia se disting entități ce au o existență continuă şi rezistă în timp, păstrându-şi identitatea chiar dacă suferă anumite schimbări (entități contigue, persistente). În al doilea rând, BFO identifică o perspectivă conform căreia în lume există entități trecătoare, reprezentate de procese, evenimente, activități, schimbări. Aceste entități au patru dimensiuni, se petrec în timp, întinzându-se pe o anumită perioadă. Cele două perspective asupra modului de existență temporală sunt surprinse de două tipuri de ontologii, una pentru entități continue şi cealaltă pentru entități trecătoare, denumite respectiv SNAP şi SPAN [16].

Abordarea ontologică poate întâmpina trei potențiale probleme: în primul rând, nu este clar cum nişte concepte ontologice pot fi găsite şi convenite de o anumită comunitate; în al doilea rând, chiar dacă ontologia a fost adoptată, nu este sigur că aceasta se va potrivi modului în care

utilizatorii datelor percep respectivul domeniu; în al treilea rând, o suită amplă de inițiative urmăreşte crearea posibilității ca ontologiile să fie accesibile programatic, prin software, există astfel pericolul de a confunda aspectele semantice cu aspectele de implementare, de procesare [17].

4. Problema unei ontologii în domeniul alertării

O motivație suficientă a necesității unui limbaj comun în domeniul alertării în situații de urgență, ce justifică demersul de cercetare, nu este explicată altfel decât se explică necesitatea standardizării în orice alt domeniu al activității umane ce implică interacțiune. O motivație suplimentară este dată de implementarea nivelului de corelare în structura sistemelor de avertizare în situații de urgență.

Astfel, în cazul în care corelarea se face direct de către actorii umani, personalul fiecărei agenții cu rol în situațiile de urgență (pompieri, ambulanță, poliție etc.) posedă, pe de o parte, un bagaj de cunoştințe particular iar pe de altă parte, o cultură organizațională specifică. Atât volumul mare de informații cât şi modul diferit de a lucra pot crea probleme de interoperabilitate şi implicit de corelare.

În cazul în care corelarea alertelor se face folosind suplimentar ajutorul calculatoarelor (şi aceasta este modalitatea propusă), probleme de interoperabilitate apar în continuare. Pentru ca două depozite de date să fie compatibile sau interoperabile, este esențial ca acestea să folosească aceeaşi terminologie, definiții etc. sau un set explicit de instrucțiuni pentru translatarea terminologiilor dintr-un limbaj în celălalt. În al doilea rând, calculatoarele execută procesări exclusiv pe baza datelor de intrare. Date de intrare inconsistente sau incoerente nu vor conduce la raționamente sau vor conduce la raționamente greşite. În al treilea rând, calculatoarele nu posedă în mod nativ abilități de apreciere a caracterului plauzibil al datelor [18].

4.1 Caracteristici ale alertelor

În ceea ce priveşte avertizarea în situații de urgență, alertele au o puternică latură contextuală, din mai multe puncte de vedere. Se pot identifica contextele locativ, temporal, operativ per ansamblu, al tipului de emițător, condițiilor particulare ale destinatarului etc. De exemplu, o alertă de incendiu are o anumită semnificație când se produce într-o clădire obişnuită şi o cu totul altă semnificație când se produce la o benzinărie sau întreprindere chimică; o alertă pe timpul nopții diferă de una pe timpul zilei; forma unei alerte care este trimisă unei persoane cu dizabilități diferă de forma unei alerte trimise unei persoane fără dizabilități şi exemplele pot continua. Ceea ce este necesar pentru cazul alertării este crearea unei ontologii domeniu, pornind de la principiile unei ontologii formale, utilă atât pentru a pune ordine în spațiul terminologic cât şi pentru procesul de corelare a alertelor, permițând raționamente umane şi raționamente maşină.

4.2 Identificarea etapelor construirii ontologiei alertării în situațiile de urgență

În construirea unei ontologii pentru avertizarea în situații de urgență se identifică, conform principiilor BFO, două etape principale: proiectarea de nivel înalt, abstract, folosind ontologia formală, rațională BFO; proiectarea de nivel scăzut, concret, folosind cunoştințe specifice aflate în posesia experților din domeniu. Aceste etape principale conțin la rândul lor alte sub-etape, unele dintre acestea care se întrepătrund. În cele ce urmează vom urmări aceste etape propuse în *Eric Little* având în vedere sarcina de a crea o ontologie pentru avertizarea în situații de urgență.

4.2.1 Crearea unui vocabular suficient de larg şi reprezentativ

Un prim pas în crearea unei ontologii este definirea termenilor şi elementelor spațiului de nume în care operează ontologia. Pentru domeniul alertării în situații de urgență, putem extrage într-o primă aproximație o serie de termeni cuprinşi de standardele de alertare existente în

prezent, cum sunt CAP, EDXL DE, TSO precum şi din alte documente de lucru specifice. Trebuie ţinut cont de faptul că diferite surse pot defini aceiaşi termini în moduri diferite.

În continuare termenii obţinuţi trebuie investigaţi, pentru a găsi eventualele sub-categorii şi relaţiile în care aceştia participă. Această sarcină ţine de evaluarea semanticii domeniului şi trebuie realizată de specialişti pe baza cunoştinţelor deţinute. În această fază utilizarea de unelte informatice la baza evaluării de termeni şi găsirii unor relaţii este puţin fezabilă. De exemplu, considerând termenul *alertă*, vor trebui investigate diferitele tipuri de alerte corespunzătoare tipurilor de dezastre: accidente, avarii, explozii şi incendii în industrie, accidente chimice cu implicaţii în afara amplasamentului, accidente chimice cu implicaţii în amplasament, accidente de construcţii hidrotehnice, accidente majore pe căile de transport, accidente nucleare şi/sau urgenţe radiologice, alunecări de teren, alunecări si prăbuşiri de teren la exploatări miniere, avarierea gravă a sistemelor de gospodărie comunală, avarii grave la magistralele de transport gaze, produse petroliere, energie electrică, avarii grave la sistemele de comunicaţii si informatică, căderi de grindină, căderi de obiecte din atmosferă şi din cosmos, căderi masive de zăpadă, contaminări de culturi vegetale, contaminări de produse vegetale şi animale, cutremure puternice, epidemii, epizootii, evenimente publice de amploare care pot genera situaţii de urgenţă, explozii necontrolate ale muniţiei rămase din timpul conflictelor militare, fenomene meteorologice periculoase, incendii, incendii la fondul forestier, invazii de dăunători, poluări ale cursurilor de apă interioare, poluări marine în zona costieră, secetă. Alte delimitări ce trebuie definite sunt date de locul, timpul alertei, emiţător, destinatari, canale de transmisie. Vocabularul rezultat va avea cu siguranţă mii de termeni.

4.2.2 Dezvoltarea categoriilor de nivel înalt (abstracte)

Ontologia de nivel înalt ce propunem să fie folosită în adoptarea unei ontologii pentru alertarea în situaţii de urgenţă este BFO, cu cele două categorii ortogonale, SNAP şi SPAN. După cum am precizat şi mai sus, SNAP surprinde instantanee ale lumii reale, care nu se întind pe o perioadă determinată de timp. În schimb, entităţilor SPAN le corespund intervale de timp în care acestea se desfăşoară. Deoarece dezastrele sunt evenimente ce se întind pe o perioadă de timp, iar alertarea în situaţii de urgenţă are caracteristicile unui proces, o ontologie pentru avertizarea în situaţii de urgenţă trebuie să cuprindă elemente ocurente SPAN; de asemenea, în interiorul proceselor vom regăsi elemente continue, obiecte SNAP. BFO identifică trei categorii mari de elemente continue: elemente continue dependente, elemente continue independente şi regiuni spaţiale, fiecare dintre acestea având la rândul lor alte sub-categorii.

4.2.3 Dezvoltarea categoriilor de nivel jos (specifice domeniului)

În această etapă elementele vocabularului pentru ontologia materială a alertării sunt mapate peste conceptele definite de ontologia formală folosită. De exemplu, „Ordinul numărul 1259 pe 2006 pentru aprobarea Normelor privind organizarea şi asigurarea activităţii de înştiinţare, avertizare, prealarmare şi alarmare în situaţii de urgenţă", ce poate fi folosit între documentele din care să se extragă lexicul pentru ontologia în discuţie, cuprinde la Articolul 6, alineatul 1, definiţia pentru *Alarmarea populaţiei*: reprezintă activitatea de transmitere a mesajelor despre iminenţa producerii unor dezastre sau a unui atac aerian şi se realizează de către autorităţile administraţiei publice centrale ori locale, după caz, prin mijlocele de alarmare prevăzute la articolul 26, pe baza înştiinţării de la structurile abilitate. Definiţia de mai sus cuprinde anumite elemente, activităţi, metode legate de alarmarea populaţiei în situaţii de urgenţă: mesajele, dezastrele, autorităţile, mijloacele etc., toate acestea necesitând a fi modelate.

4.2.4 Reprezentarea relaţiilor formale între termeni/categorii.

Legătura dintre formele de nivel înalt şi elementele materiale specifice domeniului se realizează prin integrarea manuală a categoriilor superioare ale ontologiei cu categoriile de nivel jos corespunzătoare. Spre exemplu, o alertă de inundaţie este o sub-categorie a unei alerte în general, care în continuare este o sub-categorie a entităţilor continue dependente, ce reprezintă

mai departe o sub-categorie a entităților continue, care, în fine, este o sub-categorie a entităților generice. Pentru a reprezenta aceste relații şi alte elemente ale unei ontologii, au fost proiectate limbaje specializate (Topic maps, RDF, DAML, OIL, OWL etc).

4.2.5 Dezvoltarea unui cadru computațional capabil să cuprindă ontologia domeniu

Unul dintre limbajele specializate în exprimarea ontologiilor este OWL (Web Ontology Language – a se ignora inversiunea din acronim). OWL versiunea 2 este un standard W3C, din categoria Web-ului Semantic, proiectat pentru a reprezenta cunoştințe complexe şi bogate despre lucruri, grupuri de lucruri şi relații între lucruri. OWL este un limbaj bazat pe logica computațională astfel încât cunoştințele exprimate în OWL pot fi raționate de programele de calculator, fie pentru a verifica consistența acelor cunoştințe, fie pentru a transforma cunoştințele implicite în cunoştințe explicite [19]. Există mai multe sintaxe în care OWL poate fi exprimat. Sintaxa RDF/XML [20] (Resourse Description Framework / Exchange Markup Language) este necesar să fie suportată de toate uneltele OWL. Cu ajutorul acestei sintaxe putem descrie ontologia pentru alertarea în situațiile de urgență.

```
   <owl:Class rdf:about="&snap;Alerta">
            <rdfs:subClassOf
rdf:resource="&snap;DependentContinuant"/>
   ...
   </owl:Class>

   <owl:Class rdf:about="&snap;AlertaInundatie">
            <rdfs:subClassOf rdf:resource="&snap;Alerta"/>
   ...
   </owl:Class>
```

Clasele superioare clasei DependentContinuant sunt modelate de BFO.

Pentru manipularea programatică a claselor precum cele de mai sus, o soluție este maparea acestora peste clasele unui limbaj de programare cum ar fi JAVA, astfel încât o clasă JAVA să reprezinte cât mai bine o clasă OWL. Nu dorim să cădem într-una din capcanele amintite la punctul 3, anume confundarea aspectelor semantice cu cele de implementare. Suntem conştienți de diferențele între sistemele pentru descriere logică şi cele orientate pe obiecte (semantică deschisă – semantică închisă, interpretarea implicită – explicită a apartenenței la grup, mecanisme de extindere prin inferență – extindere programatică, lipsa comportamentului – comportament nativ), însă avantajele unei mapări, fie ea şi parțială, dar funcțională, sunt foarte atractive. Ideea de a mapa ontologii exprimate OWL în limbaje de programare precum JAVA nu este nouă [21], în cele ce urmează o vom adapta pentru cazul nostru.

4.2.5.1 Elemente de mapare a claselor OWL în clase JAVA

A. Modelarea claselor

A.1. Construcții de bază - Clasele

O clasă OWL poate reprezenta de exemplu intersecția sau reuniunea altor clase OWL. De asemenea un obiect OWL poate aparține simultan mai multor clase. Se ştie că în JAVA o clasă nu poate moşteni decât de la o singură altă clasă (nu există moştenirea multiplă). Pentru a implementa aceasta, translatarea în JAVA a unei clase OWL se poate realiza inițial printr-o

interfață ce va mapa numele clasei şi semnătura metodelor de accesare (setters şi getters) ale proprietăților clasei (cu ajutorul acestor metode se vor impune şi restricțiile asupra proprietăților). Ulterior se va realiza clasa corespunzătoare, ce defineşte proprietățile şi metodele de accesare a acestora. In plus, pentru a specifica restricțiile asupra proprietăților, la fiecare metodă de setare se poate ataşa un listener şi implementa metoda corespunzătoare pentru verificarea îndeplinirii constrângerilor. Clasele realizate vor reprezenta componente *JavaBeans*. O interfață particulară este `OWL: Entity`, transpusă în JAVA ca interfața abstractă `IEntity`, pe care fiecare dintre celelalte interfețe o extind.

OWL RDF/XML

```
<owl:Class rdf:about="&snap;Alerta">
</owl:Class>
```

JAVA

```
Public Interface IAlerta {
}
Public Class Alerta Implements IAlerta {
}
```

A.2. Ierarhia de clase – implementează reprezentările axiomatice din ontologie

A.2.1. Subclasa

O clasă OWL este o subclasă a altor clase OWL dacă o instanță a celei dintâi este şi o instanță a celei de-a doua şi moşteneşte toate proprietățile acesteia. Considerând de exemplu clasa OWL:

```
<owl:Class rdf:about="&snap;AlertaInundatie">
         <rdfs:subClassOf rdf:resource="&snap;Alerta"/>
</owl:Class>
```

în JAVA reprezentarea corespunzătoare este:

```
interface IAlertaInundatie extends IAlerta, IEntity{}
class Alerta implements IAlerta{}
class AlertaInundatie implements IAlertaInundatie{}
```

A.2.2. Clasa echivalentă

O clasă OWL este echivalentă cu o alta dacă instanțele lor coincid.

```
<owl:Class rdf:about="="&snap;AlertaCutremur">
   <owl:equivalentClass rdf:resource="&snap;AlertaSeism"/>
 </owl:Class>
```

axiomă reprezentată în JAVA astfel:

```
interface IACS extends IAlertaCutremur, IAlertaSeism, IEntity{}
class AlertaCutremur implements IACS{}
class AlertaSeism implements IACS{}
```

A.3. Delimitarea claselor

În OWL este posibil ca două clase să fie distincte, adică un obiect care aparține unei clase, în mod sigur nu va aparține şi celeilalte. Acest lucru se exprimă pentru două clase din ontologia alertării în situații de urgență astfel:

```
<owl:AllDisjointClasses>
   <owl:members rdf:parseType="Collection">
     <owl:Class rdf:about="&snap;AlertaCutremur"/>
     <owl:Class rdf:about="&snap;AlertaAtacTerorist"/>
   </owl:members>
</owl:AllDisjointClasses>
```

În JAVA clasele se scriu astfel:

```
interface IAlertaCutremur extends IEntity {}
interface IAlertaAtacTerorist extends IEntity{}

class AlertaCutremur implements IAlertaCutremur{}
class AlertaAtacTerorist implements IAlertaAtacTerorist{}
```

Cele două clase sunt diferite prin codul JAVA corespunzător iar un obiect declarat ca instanță a uneia dintre ele este distinct de un obiect instanță a alteia, aşadar semantica este respectată.

A.4. Construcții avansate

Clasele OWL pot participa în relații asemănătoare celor din teoria mulțimilor, putând fi astfel folosite în construirea de clase noi, pe baza celor existente. OWL conține elemente de limbaj pentru operațiilor logice *şi*, *sau*, *negație,* iar termenii efectivi sunt cei din teoria mulțimilor: *intersection*, *union* şi *complement*. Aceşti constructori combină clase atomice (clase denumite) în clase complexe.

A.4.1. Intersecție

Intersecția a două clase OWL conține exact acele instanțe care aparțin ambelor clase. Exemplul următor arată că un apel de urgență este dintre apelurile la numărul de urgență şi cele care reprezintă un caz real de urgență:

```
<owl:Class rdf:about="ApelUrgenta">
   <owl:equivalentClass>
     <owl:Class>
       <owl:intersectionOf rdf:parseType="Collection">
         <owl:Class rdf:about="Apel112"/>
         <owl:Class rdf:about="ApelJustificat"/>
       </owl:intersectionOf>
     </owl:Class>
   </owl:equivalentClass>
 </owl:Class>
```

În JAVA această intersecție se poate reprezenta astfel:

```
interface IApelUrgenta extends IApel112, IApelJustificat, IEntity{}
class Apel112 implements IApel112{}
class ApelJustificat implements IApelJustificat{}
class ApelUrgenta implements IApelUrgenta{}
```

A.4.2. Reuniune

Reuniunea a două clase conține toate instanțele ce fac parte din cel puțin una dintre clase.

```
<owl:Class rdf:about="AgentieUrgenta">
   <owl:equivalentClass>
     <owl:Class>
       <owl:unionOf rdf:parseType="Collection">
         <owl:Class rdf:about="Pompieri"/>
         <owl:Class rdf:about="Politie"/>
      <owl:Class rdf:about="Ambulanta"/>
       …
       </owl:unionOf>
     </owl:Class>
   </owl:equivalentClass>
 </owl:Class>
```

Codul JAVA corespunzător:

```
interface IPompieri extends IAgentieUrgenta, IEntity{}
interface IPolitie extends IAgentieUrgenta, IEntity{}
interface IAmbulanta extends IAgentieUrgenta, IEntity{}
class Pompieri implements IPompieri{}
```

```
class Ambulanta implements IAmbulanta{}
class Politie implements IPolitie{}
```

A.4.3. Complement

Complementul unei clase corespunde negației logice: se compune din exact acele obiecte care nu sunt membrii ai clasei respective.

```
<owl:Class rdf:about="CladireAvariata">
   <owl:equivalentClass>
     <owl:Class>
       <owl:intersectionOf rdf:parseType="Collection">
         <owl:Class rdf:about="Cladire"/>
         <owl:Class>
           <owl:complementOf rdf:resource="CladireIntacta"/>
         </owl:Class>
       </owl:intersectionOf>
     </owl:Class>
   </owl:equivalentClass>
 </owl:Class>
```

În JAVA această distincție este realizată prin declararea separată a claselor şi interfețelor.

B. Modelarea proprietăților

În OWL obiectele pot fi descrise în primă instanță prin apartenența la o anumită clasă (şi deci implicit prin modul în care clasele sunt relaționate, tocmai pe baza obiectelor pe care le conțin). O altă metodă de descriere este prin specificarea modului de relaționare cu alte obiecte particulare. Proprietățile obiectuale OWL sunt construcții prin intermediul cărora se poate specifica în ce relații se află anumite obiecte. O altă categorie de proprietăți relaționează obiecte cu valori de un anumit tip, acestea numindu-se proprietăți tip de date. O proprietate are un domeniu (*domain*) ce specifică mulțimea obiectelor subiect logic ale proprietății şi un codomeniu (*range*) ce specifică mulțimea obiectelor predicat logic. Astfel proprietățile OWL tip de date pot fi mapate în variabile JAVA iar proprietățile OWL de tip obiect în variabile obiect de tipul clasei codomeniu. Pentru a permite cardinalitatea multiplă, variabilele vor fi de un tip *Array* sau colecție (*Set*, *List* etc.).

Proprietățile OWL pot fi supuse unei serii de restricții, începând de la domeniu şi codomeniu, valorile posibile şi terminând cu cardinalitate şi proprietăți logice. Pentru a implementa în JAVA restricțiile asupra unor proprietăți, o tehnică specifică JavaBeans este folosirea listenere-lor capabile de a aproba o modificare (*VetoableChangeListener).*

Tiparul unei astfel de implementări este următorul:

```
// Inregistrarea evenimentelor de schimbare a proprietatii in bean
bean.addVetoableChangeListener(new MyVetoableChangeListener());
```

```
class MyVetoableChangeListener implements VetoableChangeListener {
    // Acesta metoda este apelata de fiecare data cand valoarea proprietatii se schimba
    public void vetoableChange(PropertyChangeEvent evt) throws PropertyVetoException {
        // Obtinerea valorii vechi a proprietatii
        Object oldValue = evt.getOldValue();

        // Obtinerea valorii noi a proprietatii
        Object newValue = evt.getNewValue();

        // Evaluarea deciziei daca schimbarea se va aproba
        boolean veto = false;
        if (veto) {
            throw new PropertyVetoException("Motivul refuzului schimbarii.", evt);
        }
    }
}
```

4.2.6 Evaluarea ontologiei

Ultimul pas în crearea unei ontologii, conform metodologiei folosite, este evaluarea acesteia. Acest pas implică existența unui mecanism elaborat de feed-back, asemănător cu mecanismele de feedback întâlnite în modelele ciclului de viață din celelalte zone ale ingineriei sistemelor. În cazul nostru, eficacitatea ontologiei va putea fi evaluată în momentul în care aceasta va face parte integrantă din sistemul de avertizare pentru situații de urgență.

5. Extinderea în timp a ontologiei

Elementele ontologiei alertării, aşa cum au fost ele exemplificate până acum, reprezintă entități SNAP (nu au coordonată temporală internă). O ontologie a avertizării în situații de urgență este însă o ontologie dinamică, căreia nu îi poate lipsi dimensiunea temporală. BFO (ontologia de la care pornim), recunoaşte diferența dintre entitățile SNAP şi entitățile SPAN, două perspective de bază asupra lumii înconjurătoare. Pornind de la clasele SPAN BFO, se pot forma clasele unei ontologii a alertării. Spre exemplu, pentru a descrie o inundație, o vom considera ca subclasă a unui proces BFO:

```
<owl:Class rdf:about="&span;Inundatie">
   <rdfs:subClassOf rdf:resource="&span;Process"/>
</owl:Class>
```

Pentru a descrie durata procesului de inundație, putem folosi construcții definite în ontologia OWL W3C pentru concepte temporale, OWL-Time (anterioara DAML-Time) [22]:

```
<rdf:Property rdf:ID="&span;DurataInundatie">
```

```
<rdfs:domain rdf:resource="&span;Inundatie"/>
<rdfs:range rdf:resource="http://www.w3.org/2006/time#Interval"/>
</rdf:Property>
```

Translatarea în JAVA a acestor elemente se poate face după principiile descrise mai sus.

6. Concluzii şi direcţii de urmat

Interoperabilitatea între sistemele informatice particulare, cu funcţionalităţi adaptate, ce deservesc entităţi diferite, este un numitor comun care odată atins permite integrarea şi funcţionarea armonioasă ca întreg. În domeniul Sistemelor de avertizare pentru situaţii de urgenţă, corelarea alertelor este un nivel pe care l-am propus spre a completa structura clasică de monitorizare, interpretare şi alertare. Dacă perspectiva operativă asociată unei situaţii de urgenţă ar putea fi mapată într-o ontologie cât mai completă, se vor facilita atât mecanismele de judecată umană, cât mai ales mecanismele de judecată automată, realizată de calculator, ce ar aduce o sporire a gradului de încredere asociat procesului de alertare. În acest articol am prezentat premisele unei ontologii pentru avertizarea în situaţii de urgenţă, ce ar putea fi dezvoltată şi apoi adoptată de aplicaţiile software folosite în astfel de situaţii. Ducerea la bun sfârşit a acestei sarcini presupune un efort concertat din partea celor cu rol de emiţători ai alertelor în situaţii de urgenţă, autorităţilor implicate în intervenţie, destinatarilor alertelor (instituţii, populaţie) şi mediului academic. În continuare fiecare dintre etapele propuse va trebui extinsă, conform propunerilor schiţate în acest articol.

BIBLIOGRAFIE

1. **CONSTANTINESCU, N.-D.** Situaţiile de urgenţă ca provocări asimetrice la adresa securităţii lumii contemporane. Raport de cercetare. Universitatea Politehnica Bucureşti, 2010.
2. **CONSTANTINESCU, N.-D.** Suportul informatic în furnizarea parametrilor de intrare către nivelul de corelare din modelul multi-nivel al sistemelor de avertizare pentru situaţii de urgenţă, Revista „Buletin Ştiinţific", UPB, în curs de publicare.
3. **CONSTANTINESCU, N.-D.** Problema timpului în specificarea datelor de intrare pentru nivelul corelare din modelul sistemelor de avertizare pentru situaţii de urgenţă, Revista „Buletin Ştiinţific", UPB, în curs de publicare.
4. **GRUBER, T. R.** A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, 5(2):199-220, 1993.
5. **LEBBINK, H.-J.; CILIA L. M. WITTEMAN; JOHN-JULES CH. MEYE.** Ontology-Based Knowledge Acquisition for Knowledge Systems, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.8371&rep=rep1&type=pdf
6. http://www.oasis-open.org/committees/download.php/15135/emergency-CAPv1.1-Corrected_DOM.pdf
7. http://docs.oasis-open.org/emergency/edxl-de/v1.0/EDXL-DE_Spec_v1.0.pdf
8. http://www.tacticalsituationobject.org/docs/CWA_15931-1.pdf
9. **KRUCHTEN, P.; CARSON WOO; KAFUI MONU; MANDANA SOTOODEH.** A Human-Centered Conceptual Model of Disasters Affecting Critical Infrastructures. În: Proceedings of ISCRAM 2007.
10. **DI MAIO, PAOLA.** An open ontology for open source emergency response system, Proceedings of ISCRAM 2005, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.1829&rep=rep1&type=pdf.

11. **MALIZIA, A.; FRANCISCO ASTORGA-PALIZA; TERESA ONORATI; PALOMA DIAZ; IGNACIO AEDO.** Emergency Alerts for all: an ontology based approach to improve accessibility in emergency alerting systems, Proceedings of the 5th International ISCRAM Conference – Washington, DC, USA, May 2008

12. **XU, WEI; SISI ZLATANOVA.** Ontologies for Disaster Management Response, , http://www.gdmc.nl/zlatanova/thesis/html/refer/ps/WX_SZ_2007.pdf.

13. **NGO, P.; DUMINDA WIJESEKERA.** Using Ontological Information to Enhance Responder Availability in Emergency Response, George Mason University, Technical Report GMU-CS-TR-2010-13.

14. **XU, WEI; SISI ZLATANOVA.** Ontologies for Disaster Management Response, http://www.gdmc.nl/zlatanova/thesis/html/refer/ps/WX_SZ_2007.pdf

15. **LITTLE, E.** A Proposed Methodology for The Development of Application-Based FormalOntologies, Workshop on Reference Ontologies vs. Application Ontologies (2003).

16. **GRENON, P.; BARRY SMITH.** SNAP and SPAN:Towards Dynamic Spatial Ontology, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.9297.

17. **DELACAMBRE, L.; VIJAY KATHRI; YAIR VAND; BARBARA WILLIAMS; CARSON WOO; MARK ZOZULIA.** Eliciting Data Semantics via Top Down And Bottom Up Approaches: Chalenges and Oportunities, Conceptual Modeling - ER 2006, Lecture Notes in Computer Science, 2006, Volume 4215/2006, 548-551, DOI: 10.1007/11901181_41.

18. **SPEAR, A. D.** Ontology for the Twenty First Century: An Introduction with Recommendations, www.ifomis.org/bfo/documents/manual.pdf.

19. http://www.w3.org/TR/owl2-primer/

20. http://www.w3.org/TR/REC-rdf-syntax/

21. **KALYANPUR, ADITYA; DANIEL JIMÉNEZ PASTOR; STEVE BATTLE; JULIAN PADGET.** Automatic Mapping of OWL Ontologies into Java, In: Maurer, G. R. F., ed. Proceedings of Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE), Jun 2004.

22. http://www.w3.org/TR/owl-time/