Computed Ontology-based Situation Awareness of Multi-User Observations

Siska Fitrianie
Man-Machine Interaction Group,
Delft University of Technology
s.fitrianie@tudelft.nl

Leon J. M. Rothkrantz
Delft University of Technology and
Netherlands Defence Academy
l.j.m.rothkrantz@tudelft.nl

ABSTRACT

In recent years, we have developed a framework of human-computer interaction that offers recognition of various communication modalities including speech, lip movement, facial expression, handwriting/drawing, gesture, text and visual symbols. The framework allows the rapid construction of a multimodal, multi-device, and multi-user communication system within crisis management. This paper reports the approaches used in multi-user information integration (input fusion) and multimodal presentation (output fission) modules, which can be used in isolation, but also as part of the framework. The latter is able to specify and produce context-sensitive and user-tailored output combining language, speech, visual-language and graphics. These modules provide a communication channel between the system and users with different communication devices. By the employment of ontology, the system's view about the world is constructed from multi-user observations and appropriate multimodal responses are generated.

Keywords

Multimodal HCI, multi-user information retrieval, multimodal fission output.

INTRODUCTION

Analysis of past disasters - 9/11 and hurricanes Katrina and Rita by Moore (2006), points to communication as a limiting factor in disaster response. Current approaches to construct globally consistent views of such an event suffer from problems of (a) the setting of such events is constantly changing, (b) the information is likely to be distributed piecemeal across geographically distant locations; it is difficult and timely to collaborate and verify the obtained information, and (b) the intense nature of the event causes various uncertainty, e.g. errors in the spontaneous speech input under stress recognition. This results in fragmented, ambiguous, inaccurate and even irrelevant communicated data (Sharma et al., 2003). In providing appropriate information to ensure interoperability of emergency services, situation awareness and high-quality care for citizens, the ability to collect and present dynamic information in contextually and temporally correlated information are necessary. Furthermore, recent developments in technology offer the citizens as (human) observers - possibilities for more diverse computerized communication devices: workstation, laptop/tablet computer, PDA, mobile telephone. As mobility becomes ubiquitous, multimodality becomes the inherent basis of the interaction paradigm. Multimodal interfaces can offer various ways, modalities and devices for users to interact in a more natural fashion with the provided services (Oviatt et al., 2004). This advanced technology gives extra requirements, such as: (1) to accommodate the flexible switching of communication modes and devices, which can have different capabilities of the physical device and of the provided I/O modalities, (2) to adapt the application and presented information to user context variables (e.g. profile, emotion, location) which may change over time due to mobility and dynamic environment, and (3) to offer a unified view of its services under such technology constraints and dynamic changes. These requirements yield serious interface and interaction design issues.

The introduction of novel ICT in the crisis management domain can help to provide more detailed and accurate situation overviews that are current and shared amongst all management levels (Moore, 2006). Towards this goal, this paper reports our investigation on methodologies to improve multi-user information retrieval and exchanges during crisis respond activities. We believe that accurate and easy access of information can support better situation awareness. Here, we present our approaches in: (1) merging incoming information and (2) specifying and producing context-sensitive (and user-tailored) information - by the employment of ontology from and to multi-user, multi-device and multimodal. The latest includes allocating and coordinating information across media, i.e. typed or spoken language, visual language, and graphics.

RELATED WORK

Years after 9/11, 2001, efforts to develop technology in crisis management emphasize the development of more sophisticated ICT-based emergency support systems, e.g. a web-based reporting interface in *CAMAS* (Mehrotra et al., 2004) and VCMC (Otten et al., 2004), an icon-based interface for sharing and merging topological maps in damaged buildings (Tatomir & Rothkrantz, 2006) and a multimodal framework that facilitates decision making in control rooms (XISM - Sharma et al., 2003).

The information presentation (in a multimodal system) often is framed into: deciding what to communicate and how to communicate it. Some systems use an ontological language for encoding features of modality processing and planning for information presentation, e.g. M3L in SmartKom (Wahlster, 2006) and RDF-OWL in COMIC (Foster, 2004), others use simple XML (Match (Johnston et al., 2002) and MACK (Cassell et al., 2002)). SmartKom utilizes an *overlay operation* (Alexandersson & Becker, 2003) to integrate user inputs, while COMIC translates them into logical forms in a variation of Augmented Transition Network containing all possible dialogue moves. Most multimodal systems are designed for single-user services. A presentation planner retrieves the unified information and specifies presentation for language, graphics and animation generation. Information about user, system beliefs, dialogue history and communication device constraints are considered in all processes. The language generation typically uses some linguistics approaches, such as Tree Adjoining Grammar (Joshi & Vijay-Shanker, 1999) in SmartKom, n-grams and Combinatory Categorial Grammar (Steedman & Baldridge, 2005) in COMIC, and template-based in Match. Techniques like stack in Match, schema-based in COMIC and XIMS, and rule-based in MACK, are applied for coordinating virtual agents' behaviors, graphical displays and speech output.

OVERVIEW: COMPUTATIONAL HUMAN INTERACTION MODELING FRAMEWORK

A framework that allows the rapid construction and evaluation of multimodal human-computer interaction (HCI) systems has been developed (Fitrianie et al., 2007). The development aims at modules integration that is independent of the availability of modalities, therefore, a HCI system can be constructed to support communication between different actors via different devices to work collaboratively resolving crisis situations. Currently, a demonstrator system for reporting observations is developed, which is able to collect up to date observations, interpret them automatically and form a global view about the reported events.

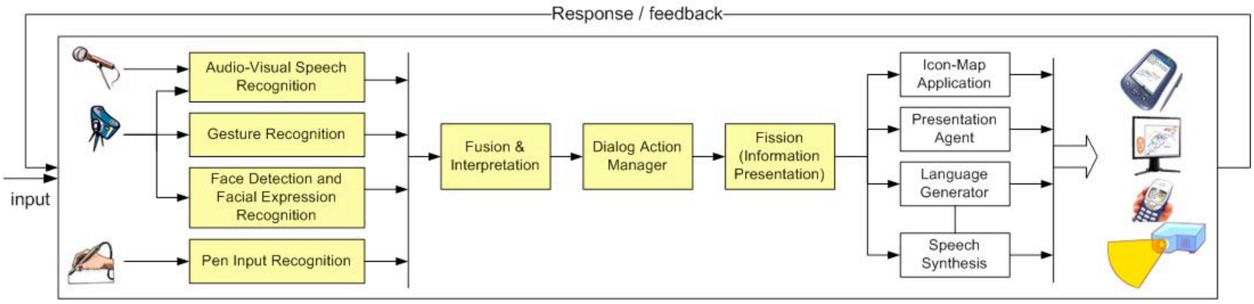
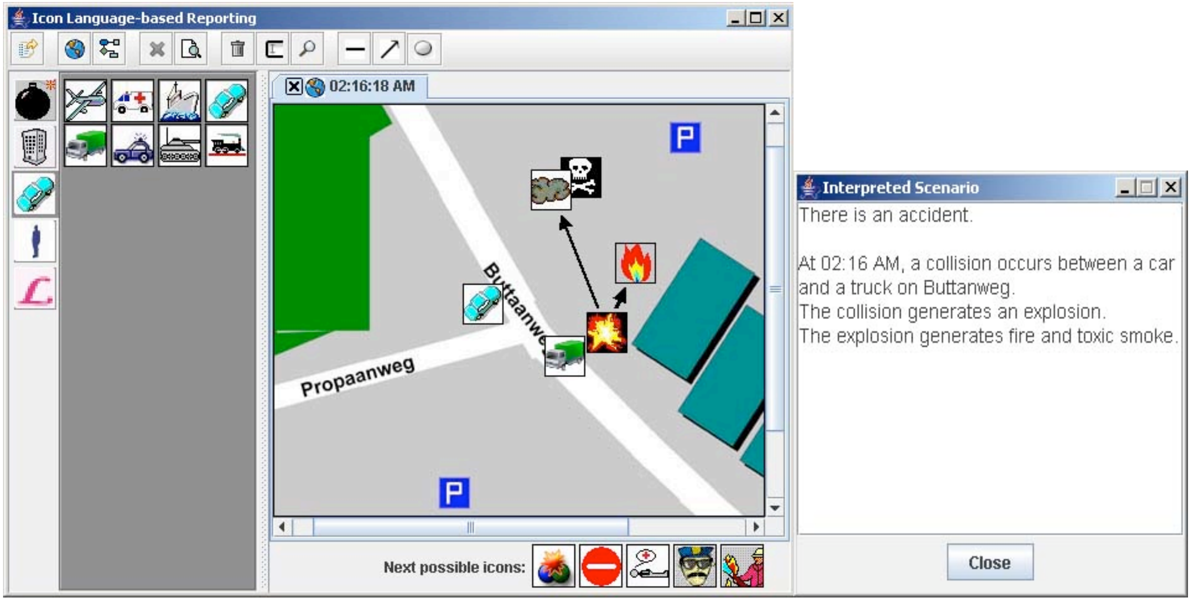


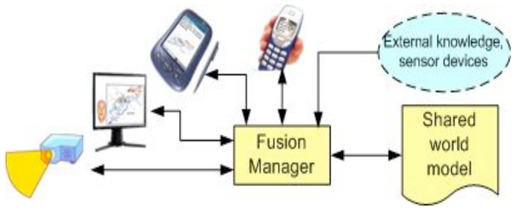
Figure 1 The framework architecture for single user

The framework includes input recognition modules of different modalities: text, speech, visual language, gesture, pen writing/drawing, and facial expression (Figure1). The output combines text, synthesized speech, visual language and control of the underlying user interface. A presentation agent that is able to generate speech with intonation and facial behaviors is also utilized. A map interface for supporting people with geospatial information was designed (Figure 2(a) – Fitrianie et al., 2008). It provides drawing tools for a free and natural way of describing a crisis situation using predefined icons, lines, arrows, and shapes. Icon-strings, text and photos are offered for non-spatial inputs. A coherent and context dependent interpretation and textual crisis scenario of the inputs are constructed as feedback to the user input. The icons represent objects and events in the world (e.g. fire, car, smoke). This representation supports faster interaction (Kjeldskov & Kolbe, 2002), reduces the ambiguity of the communicated information (Norman, 1993), and provides a language independent message construction (Perlovsky, 1999).

The framework is designed to support various roles within the crisis management, including professionals and civilians in the field and control room (Figure 2(b)). A centralized Fusion Manager processes and integrates every newly reported situation from all users and adapts the world view accordingly then sends it back to the network and shares with the users. All modules are integrated in an *ad hoc* fashion using the iROS middleware system (Johansson, 2002).



(a)



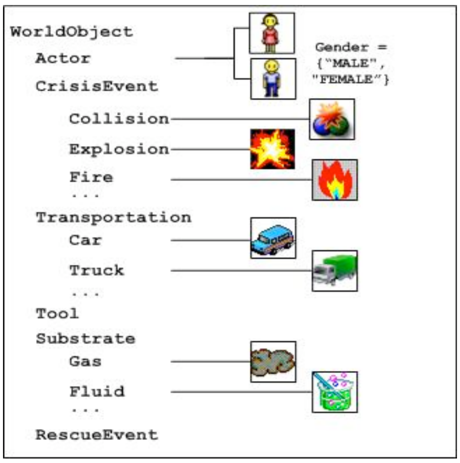
(b)

Figure 2 (a) Icon-language-based application for reporting situations and (b) a schematic overview of the communication system: multimodal, multi-device and multi-user

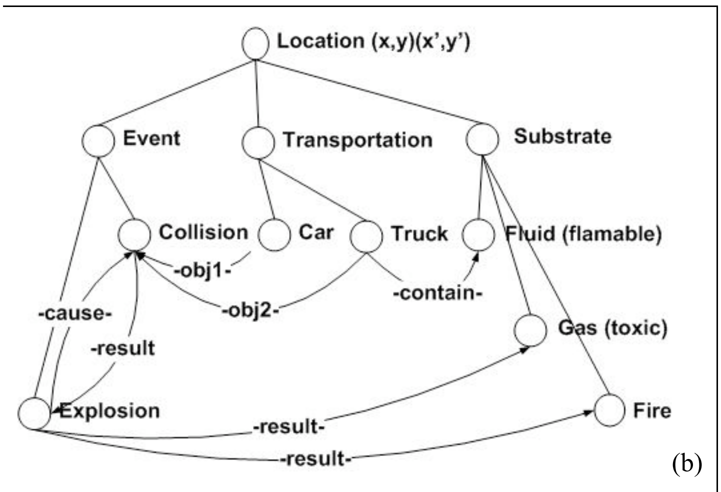
The knowledge representation used in the framework is introduced in the next section. Further, we will focus on our approaches utilized in the fusion and fission modules for integrating and presenting information. We continue with a brief description of our user tests. Finally, we will conclude our contributions.

ONTOLOGY AS KNOWLEDGE REPRESENTATION OF THE WORLD, THE USER AND THE TASK

To support consistent communication data inter-components, knowledge representation about the world, the user, and the task in our framework are defined as ontology in RDF-OWL.



(a)

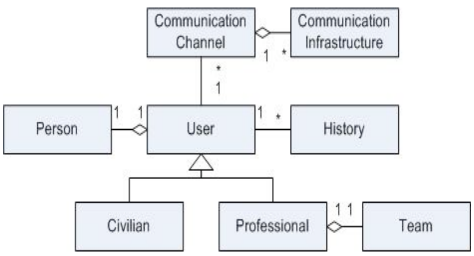


(b)

Figure 3 (a) The WorldObject class Taxonomy; the icons are the instances of a class e.g. the icon "smoke" is a subclass of "Gas", and (b) a graph-based representation of "a Collision of a Car and a Truck result in an Explosion; the explosion causes toxic Gas and Fire"

A world model is a representation of two geo-referenced contexts: (1) a chain of temporal specific events and a group of dynamic objects in action at a certain location in the world and (2) geographical information concerning the crisis location, such as buildings, streets and parcels. It has direct links conceptually to the icons on the user interface (Figure 3(a)). The geospatial model of crisis situations is described using graphs. A graph connects its nodes based on their approximated world spatial coordinates. The nodes represent objects, actions or events in the world. They contain specific information including current status (e.g. living condition, spatial state), temporal information (e.g. frequency, timestamp), and spatial information (e.g. current location, origin, destination, path). The arcs represent the hierarchy of individuals and the arrows show the relations among nodes (e.g. result, cause and contain). At the root, a node describes the perspective location of the event (see illustration in Figure 3(b)).

A user model refers to a dynamic model of registered users (Figure 4(a)). It contains their identity, emotional states, interaction time, current communication media and modalities, and current location in the world. Two types of users (civilian and professional) can be a part of actors in the world model and referred by their id. A Person stores most static information about a user, while the dynamic information is in the User and archived in the History. One or more Communication Channels can be registered by a user; a channel can have one or more Communication Infrastructures, which contain information about available media and modalities.



(a)

```
…
<dialogueActType>statement</dialogueActType>
<priority><value>1</value></priority>
<communicationchannel>
  <communicationinfrastructure>
    <mode>short-message</mode><mediatype>text</mediatype>
    …
  <communicationInfrastructure>
  …
</communicationchannel>
<concept>
  <name>Escort</name>
  <property>
    <name>agent</name>
    <value><instance id="WM01Police01" type="Police"/></value>
  </property>
  <property>
    <name>patient</name>
    <value><instance id="WM01Paramedic01" type="Paramedic"/></value>
  </property>
  <property><name>source</name>…</property>
  <property><name>destination</name>…</property>
</concept>
…
```

(b)

Figure 4 (a) User model class diagram and (b) an example of a dialogue act: "informing a user that a policeman is going to escort the paramedic from a source location to a destination"

In our framework, a dialogue action manager (DAM) controls HCI flows. It defines dialogue acts and selects output modalities to convey the message (Figure 4(b)). Currently, nine dialogue acts are implemented: (1) statement: to notify an event, (2) request: to command an action, (3) ask: to request information, (4) confirmation: to clarify information, (5) acknowledge: to display information, (6) highlight: to foreground information on the display, (7) removal: to withdraw information from the display, and (8) affirmation or (9) negation: to notify the system's agreement on user's action. In the ontology, these acts are defined including its priority, option modalities, schedule, references to other processes and the world model.

MODEL OF COMPUTED SITUATION AWARENESS

In crisis situations, knowing the situational information about the current world is the precondition for setting goals and domains of actions. It relies on observation reports from all parties involved. The observations can be from people in the field and sensor devices, e.g. smart cameras, smoke detectors and wind meters (Figure 5).

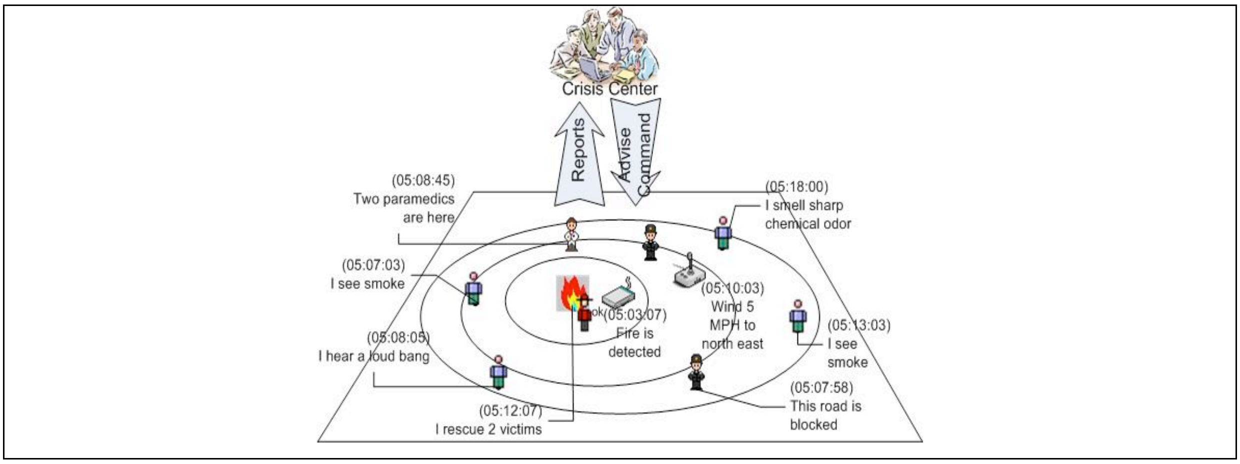


Figure 5 Disaster dispersion (ellipses) occurs on the world; agents in the field report their observation

In our framework, all actors send their observations using a certain communication device. Human actors can create messages on its interface using available modalities. The fusion module in a single-user interface builds concepts from various preprocessed input modalities, which are supplied by all the available input recognition modules. The result of this process is a contextually and temporally structure of concepts. This is handed over to the DAM and may assist it in forming feedback to the user. The DAM sends the single user's processed message to the (multi-user) Fusion Manager. The Fusion Manager attempts to relate the existing structure of the current system's world model with the new message and then broadcasts the latest world knowledge. Together with a selected dialogue act from the DAM, the fission module of a single user interface adapts the current world model and produces a user-tailored information presentation. All users receive the most up-to-date information about the current situations on their communication interface. Figure 6 shows a simplified view of these processing flows. Ontology and scripts are utilized in all processes.

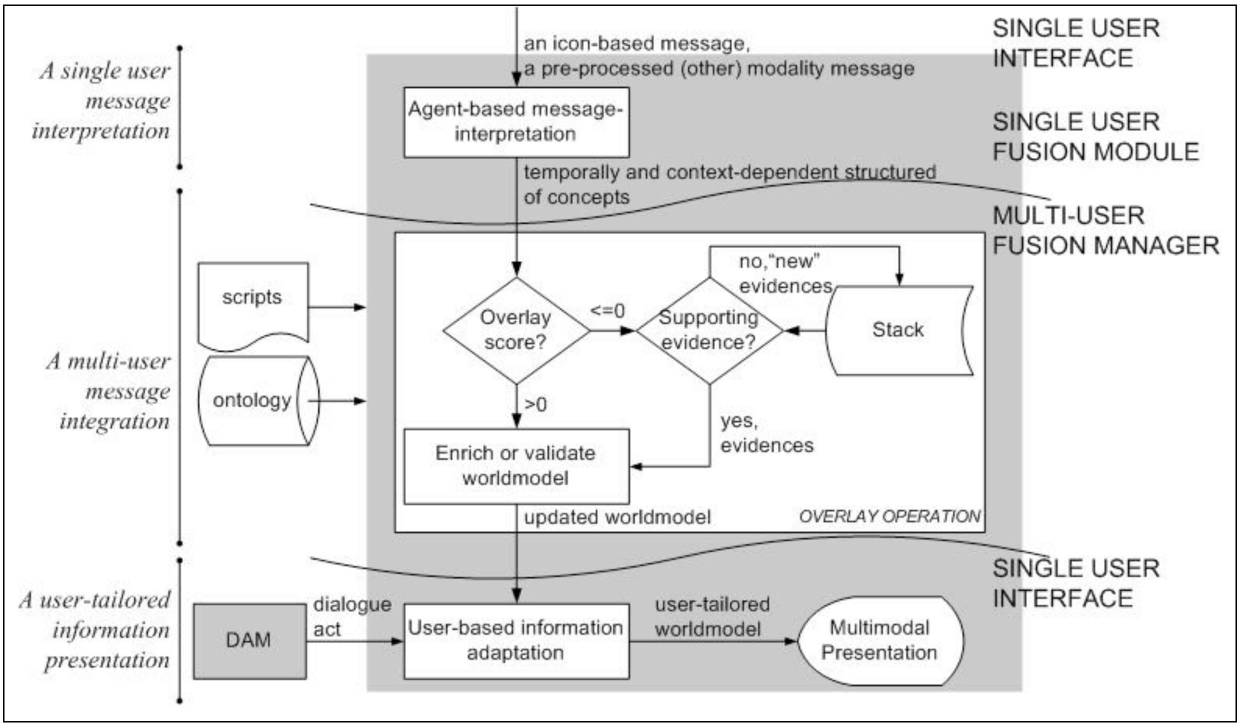
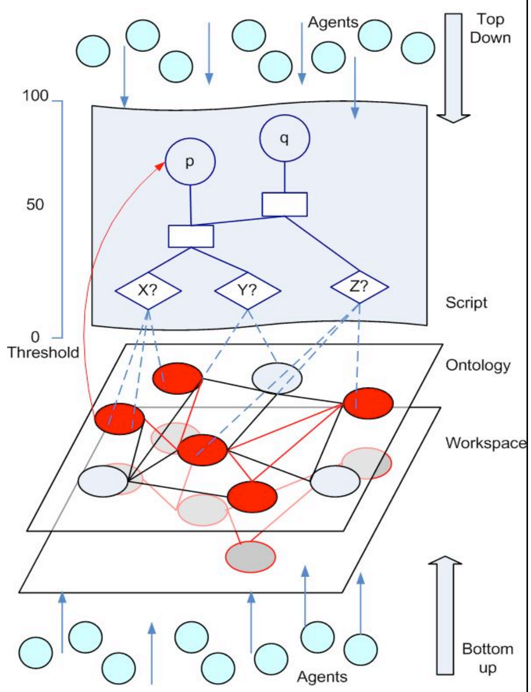


Figure 6 A schematic flowchart of proposed approaches in processing multi-user messages

A Single User Message Interpretation

In Fitrianie et al., (2008), an agent-based architecture for constructing a coherence and context dependent interpretation of a (preprocessed) multimodal user input has been discussed (Figure 7(a)). Inspired by Dor (2005), the interpretation is derived from purposive behavior emerging from interaction between multiple concepts on a user *workspace*. *Ontology* holds predefined concepts in the form of a graph of concepts and properties. As *agents* find matching instances in the workspace (bottom up), they activate their corresponding concept-property in the ontology and dynamically build relationships among these concepts and message structures. When a concept in the ontology is activated, it will try to assign as many properties as possible by launching more agents to search for more values. As a result, an active concept will spread some activation to other relevant concepts. Furthermore, an active concept can activate one or more *scripts* (top-down).

Based on Schank & Abelson (1977), a script represents the chain of events that identify a possible crisis scenario (Figure 7(b)). An active script can launch more agents to evaluate its conditions. As more concepts are active in the ontology, more key concepts are represented most common to some competing scripts. As a result, certain scripts become impossible and are removed from competition. This process continues until, there is only one script left. At this point, it is assumed that an enough and coherence structure of the communicated message interpretation is produced.



(a)

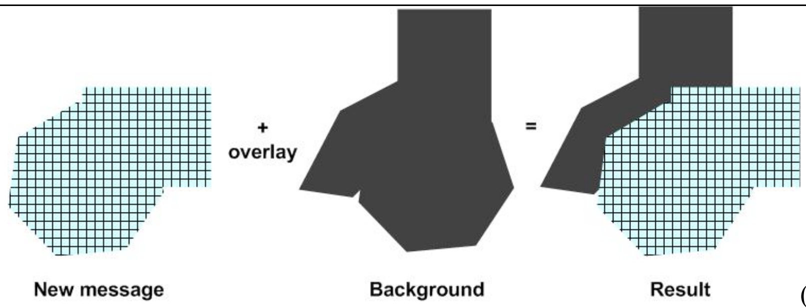
```xml
<script name="accident" type="acknowledgement">
 <frame name="number" logic="and" weight="60">
  <condition operator="biggerThan" function="number">
   <concept name="Transportation" propertyName="number"/>
   <constant value="1" type="int"/>
  </condition>
  <condition operator="biggerEqualThan" function="number">
   <concept name="Victim" propertyName="number"/>
   <constant value="1" type="int"/>
  </condition>
 </frame>
 <frame name="location" logic="and" weight="1">
  <condition operator="lessEqualThan" unit="meter">
   <condition function="distant">
    <concept name="Transportation" propertyName="location"/>
    <concept name="Road" propertyName="Topology"/>
   </condition>
  </condition>
  <condition><constant value="1" type="double"/></condition>
 </frame>
 <frame name="number" logic="and" weight="100">
  <condition operator="biggerThanEqual" function="number">
   <concept name="Collision" propertyName="number"/>
   <constant value="1" type="int"/>
  </condition>
 </frame>
  ...
</script>
```

(b)

Figure 7 (a) An agent-based interpretation for a single multimodal message and (b) an example of a script (in XML)

A Multi-User Message Integration

To form an aggregated world view based on observations of multiple users, *overlay* operation is utilized (Alexandersson & Becker, 2003). It is a formal operation based on unification of typed feature structures – TFS (Carpenter, 1992 – Figure 8(b)). Overlay can be seen as putting two typed feature structure - covering and background - on top of each other (Figure 8(a)). The background can be viewed as the old state of the world, while the covering is an incoming message.



(a)

$$\begin{bmatrix} t \\ a_1 : f_1 \\ \ldots \\ a_n : f_n \end{bmatrix}$$

(b)

Figure 8 (a) Schematic view of overlay operation and (b) graphical representation for denoting a TFS

All inputs are aggregated in the following:

1. Map the RDF-OWL document of a new message into TFS, as follows:
   - collect all active concepts,
   - fill all properties with actual values (from the instances), and
   - if a TFS is denoted as $\{t, [a_1 : f_1, \ldots, a_n, b_n]\} \in$ TFS where t is the type, $[a_1 \ldots a_n] \in F$ the features and $[f_1 \ldots b_n] \in A \cup$ TFS the values (Figure 9), map all concepts into t_i with properties $[a_{i1} \ldots a_{in}]$, which have values $[f_{i1} \ldots b_{in}]$.

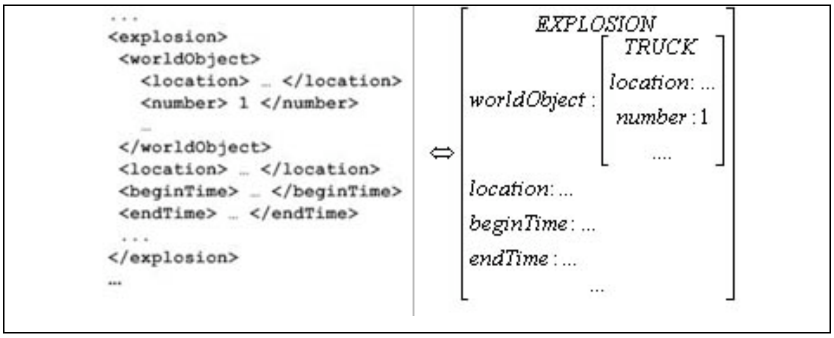


Figure 9 A set of active concepts (in XML) and its corresponding TFS

2. Assimilate the background to the covering. Here, predefined margin timestamp and location thresholds of an active concept are defined and applied in the operation. The assimilation is necessary to ensure that the features in the background defined in the least upper bound type (*lub*) of the types of the covering and background. The assimilation of a to b ($a, b \in$ TFS) is defined as $a|b := \{t_b, [a_i : f_1, \ldots, a_j : f_j]\}$ such that $a_i = \{t_i, [a_{i1} : f_{i1}, \ldots, a_{in} : f_{in}]\}$ and $a_i \in t_s = lub\ (t_a, t_b)$. A function $lub(t_a, t_b)$ computes the most specific common super-type for two types of a and b.

3. Overlay the background b to the covering a where $a = \{t_a, [a_1 : f_1, \ldots, a_n : f_n]\}$ and $b = \{t_a, [b_1 : g_1, \ldots, b_m : g_m]\}$ and $t_b' = lub\ (a, b)$:

   $$overlay(a, b) = overlay'(a, b|_a)$$

   if $a \neq b$ then $overlay(a, b) \neq overlay(a, b)$; and if a and b are "unifiable" then $overlay(a, b) = overlay(b, a) = unify(a, b)$. A function $overlay'(a, b)$ is defined as four cases:

   [i] If recursion:
   $overlay'(a, b)=\{t_a, [c_i : h_i \mid c_i = a_j = b_k, h_i = overlay(f_i, g_k)]\}$, where $f_j, g_k \in$ TFS.
   [ii] If the covering and the background have (atomic) values:
   $overlay'(a, b)=\{t_a, [c_i : h_i \mid c_i = a_j = b_k, h_i = f_i]\}$, where $f_j, g_k \in$ A.
   [iii] If a feature is absent in the background:
   $overlay'(a, b)=\{t_a, [c_i : h_i \mid c_i = a_j, h_i = f_i, c_i \neq b_k, 1 \leq k \leq m]\}$.
   [iv] If a feature is absent or has no value in the covering:
   $overlay'(a, b)=\{t_a, [c_i : h_i \mid c_i = b_k, h_i = g_k]\}$.

4. Calculate overlay score. This score (Pfleger et al. 2002) is defined to reflect how well the covering fits the background in terms of non-conflicting features:

   $$score(co, bg, tc, cv) = \frac{co + bg - (tc + cv)}{co + bg + tc + cv} \in [-1,1]$$

   where *co* is the total number of non-conflicting or absent features for cases [i], [ii] and [iii] and values in the covering for cases [ii] and [iii]; *bg* is the total number of non-conflicting or absent features in the background for cases [i], [ii] and [iv] and values for cases [ii] and [iv]; *tc* is the total number of not-identical types of both the covering and the background; and *cv* is the total number of conflicting values (case [ii], if the value of a feature of the background is overwritten). The $score(co, bg, tc, cv) = 1$ indicates the feature structure are unifiable and $score(co, bg, tc, cv) = -1$ indicates all information from the background has been overwritten.

15:20:10 An observer reports: **"A truck is on fire on Buttanweg"**	**Initial world model ⇔** .... $\mathbf{FTS}: \begin{bmatrix} FIRE \\ causal: \begin{bmatrix} TRUCK \\ location: \begin{bmatrix} ADDRESS \\ streetName: Buttanweg \end{bmatrix} \\ number: 1 \end{bmatrix} \\ location: \ldots \\ beginTime: \ldots \end{bmatrix}$
15:21:02 A policeman reports: **"A truck is exploded on Buttanweg 321 creating fire"** $co = 14$; $bg = 12$; $tc = 1$; $cv = 7$; $score(co, bg, tc, cv) = 0.529$ **The message is validated by a script: the fire is caused due to explosion. It enriches and validates the world model.**	**updated world model ⇔** .... $\mathbf{FTS}: \begin{bmatrix} FIRE \\ causal: \begin{bmatrix} EXPLOSION \\ causal: \begin{bmatrix} TRUCK \\ location: \begin{bmatrix} ADDRESS \\ streetName: Buttanweg \\ number: 321 \end{bmatrix} \\ number: 1 \end{bmatrix} \\ location: \ldots \\ beginTime: \ldots \end{bmatrix} \\ location: \ldots \\ beginTime: \ldots \end{bmatrix}$
... a conflicting message coming: **15:22:07 An observer reports:** **"Awful smell in the air near Propaanweg"** $co = 0$; $bg = 0$; $tc = 1$; $cv = 0$; $score(co, bg, tc, cv) = -1$	**a new message (stored in the stack) ⇔** $\mathrm{FTS}: \begin{bmatrix} GAS \\ hazardousLevel: unknown \\ location: \ldots \\ beginTime: \ldots \end{bmatrix}$
... after sometime, a new evidence is coming .. **15:29:55 A fireman reports: "The truck (on Buttanweg 321) contains toxic fluid"** $co = 3$; $bg = 21$; $tc = 0$; $cv = 0$; $score(co, bg, tc, cv) = 1$ **This evidence supports the previous message. It is validated by a script: the toxic fluid in fire may release toxic gas. Both messages enrich the world model.**	**updated world model ⇔** .... $\mathbf{FTS}: \begin{bmatrix} GAS \\ causal: \begin{bmatrix} FIRE \\ causal: \begin{bmatrix} EXPLOSION \\ causal: \begin{bmatrix} TRUCK \\ location: \begin{bmatrix} ADDRESS \\ streetName: Buttanweg \\ number: 321 \end{bmatrix} \\ contain: \begin{bmatrix} FLUID \\ hazardouslevel: HIGH \\ \ldots \end{bmatrix} \\ number: 1 \end{bmatrix} \\ location: \ldots \\ beginTime: \ldots \end{bmatrix} \\ location: \\ beginTime: \ldots \end{bmatrix} \\ hazardouslevel: HIGH\ldots \\ beginTime: \ldots \\ location: \ldots \\ \ldots \end{bmatrix}$
15:29:57 A fireman reports: "Toxic smoke is discovered ... " **...**	**A Smoke is a subclass of a Gas. This message will validate the world model.** **...**

Figure 10 Applying overlay on observation reports

5. Enrich or validate the state of the world (see examples in Figure 10). The current implementation processes all messages that have the overlay score > 0. Messages with overlay score ≤ 0 are stored in a stack until some evidences support them (validated by a script). In particular, an activation level of active concepts in the ontology is defined. The level will grow if new messages include these concepts and decay at predefined intervals, otherwise. By this mechanism, only up-to-date (observed) concepts are active in the aggregated world-model. We expect that the system's knowledge of the world is built based on reliable messages; while those un-reliable messages by the mechanism eventually will be discarded.

The entire interpretation process and results do not include any private information of users, except their location. All observers are treated as (anonymous) actors doing a certain action in a certain location. The information about these actors, combining with information about specific objects and events appeared in the world, builds up a scenario of a certain crisis event, which is validated by the scripts.

A User-Tailored Information Presentation

The presentation module receives a dialogue act for a specific user from the DAM and the current world model (Figure 11). To support the rapid changes of a crisis situation, the presentation can be triggered by the change in the world model.

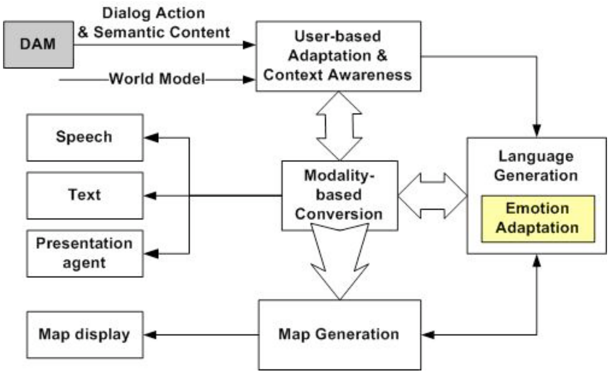


Figure 11 The architecture of the fission module

User-Based Information Adaptation

With direct access to and the availability of real time context and technical information of three knowledge sources: the user, the task and the world, the *Adaptation component* plans presentation contents based on the following user contextual information.

- *User role*. Filtering messages using predefined flags, i.e. public (for all), protected (for rescue workers and crisis room operators), and private (for a certain group or operators).
- *User location*. Locating the perspective area of the world model using a set of policies based on the current user position and models of physical phenomena including the dispersion of the crisis and its impact to other entities.
- *Available modalities*. Selecting output modalities based on the option modalities selected by the DAM and the input modalities used by the user, e.g. if the input includes gestures, the output will include a large screen display, the map interface and speech; this policy is used for operators in a crisis room.
- *Interaction time*. Presenting the most up-to-date information based on the user's dialogue history.
- *User emotion* (used by the language generation component).

Multimodal Presentation

The *Modality-based Conversion component* allocates the user-adapted information across the language generation and the map display generation component. This component divides all active concepts to be a set of sequence segments. All segments are linked and synchronized with concepts that are processed in the language generation and map display generation. The component estimates the time needed for each processed segment depending on the selected modalities. It controls the output using this timing information to create a full schedule for the turn of all segments. Each language segment is generated one by one synchronized with the display (of active or highlighted concepts). Other active concepts that are not linked to any language generation segment are displayed directly. This component also informs the display component about those concepts that are not longer active.

The *language generation component* works by a simple recognition of the dialogue act and the concept-name and a substitution of the property-names by their values controlled by a modified-AIML format. AIML (Wallace, 2003) is extended-XML specifications for language generation in dialogue systems. A category (a knowledge unit in a dialogue) is defined within a certain topic. It provides templates (of the output messages) that can be selected depending on the <concern> tag, which refers to the current user emotional state. The same concept and dialogue act can have many categories, but a different set of properties. Inside the template, <get> tags can be substituted by a value (a) from a property (e.g. `<get name = "source" type = "Location"/>`) and (b) of a function (e.g. `<get function = "getStaticObjectInfo (upperLeft, bottomUp" type = "String")`). If the "type" is a concept-name, the component will search the category with the corresponding concept recursively (Figure 12).

```
<aiml>
 <topic name="CAR ACCIDENT">
  <category>
   <dialogAct>statement</dialogAct>
   <concept name="Escort"/>
   <properties>
     <property name="patient"/><property name="source"/><property name="destination"/>
   </properties>
   <template type="short-message">
    <concern name="neutral" value=""/><random>
      <li>A <get name="agent" type="Class.name"/> will escort the
       <get name="patient" type="Class.name"/> from <get name="source" type="Location"/> to
       <get name="destination" type="Location"/>.
     </li>
     <li>A <get name="agent" type="Class.name"/> and a <get name="patient" type="Class.name"/>
      will come to <get name="destination" type="Location"/> from <get name="source" type="Location"/>
     </li>
     ...
    </random></concern>
   ...
   </template>
  </category>
  <category>
   <dialogAct>*</dialogAct>
   <concept name="Location"/>
   <properties><property name="upperLeft"/><property name="bottomUp"/></properties>
   <template type="*">
      <get function="getStaticObjectInfo (upperLeft, bottomUp)" type="String"/>
   </template>
  </category>
 ...
 </topic>
...
</aiml>

Example of resulted text:
"A police will escort a paramedic from A4 West BeneluxTunnel km 30 to Buttanweg no. 321 "
```

Figure 12 An example of the text generation from two AIML units

The *Display component* generates a representation of a map interface as follows:

- *Adding icons*. All correspondence icons of active concepts by ensuring none of the instance of the concepts are displayed more than once. A function is used to map world coordinates to screen coordinates.
- *Adding links between icons*. An arrow shows a causality (the property "cause"). A line shows relations between icons: (1) if one is the value of the other's property and (2) if they are the property values of an active concept that does not have a correspond icon, e.g. a concept "Passenger" with properties "vehicle" (type="Transportation") and "agent" (type="Actor") does not have any correspondence icon.
- *Update icons and links*. The component updates all correspondence icons (and their links) except those are specified by the user (only the user can remove them). This action can be forced by the dialog act "remove".
- *Highlight concepts or certain locations on the map*. This action is triggered by a dialog act "highlight". It also supports visually the language component presentation, e.g. the highlighted icons of "Paramedic" and "Policeman" in Figure 13 are displayed while the system synthesizes the text resulted

in Figure 12. The highlight is removed after some predefined time or forced by the dialogue act "remove".

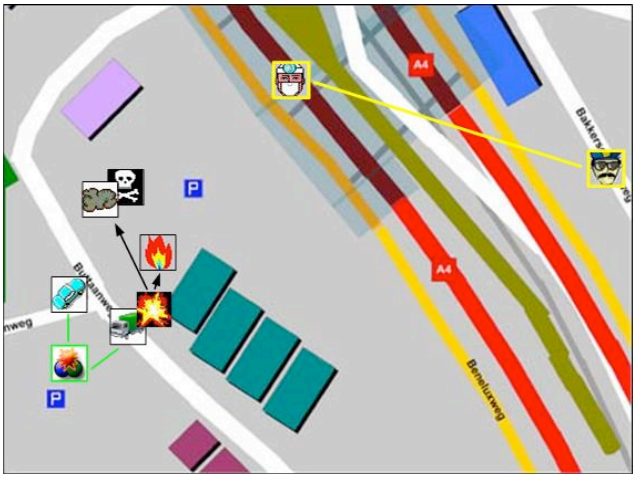


Figure 13 A display shown while informing the user that a policeman and a doctor will come to the crisis scene; the concept "Collision" (green bordered) is derived from the selected script in the message interpretation process (compare to Figure 2(a))

Generated event	Possible action
05:03 Fire at area (x,y)(x',y') A fire detector detects fire and activates alarm.	*Human observers*: see fire and report it to the crisis center. *Crisis center*: receive reports and send a call to Fireman for checking.
05:07 Fire and smoke develop A thermometer measures the current temperature. Firemen are on their way to (x,y)(x',y').	*Human observers*: report to the crisis center about developed smoke and the present of firemen. *Crisis center*: receive reports, send policemen to (x,y)(x',y').
05:08 Explosion Policemen block roads in the danger area.	*Human observers*: report to the crisis center about damage and casualties, a loud bang and the present of policemen. *Crisis center*: order extra units firemen and paramedics to (x,y)(x',y').
05:10 Professionals in action A wind meter measures the wind speed and direction. Paramedics arrive at (x,y)(x',y'). Firemen start extinguishing fire and rescuing victims, paramedics help victims and policemen guard the area.	*Human observers*: report to the crisis center about the professionals' activities. *Crisis center*: send advise to the professionals and civilians.
…	…

(a)

(b)

Figure 14 (a) First minutes-scenario based on Figure 5 and (b) example photographs of crisis situations

EXPERIMENTS

To assess our proposal, laboratory tests were performed. It also addressed our demonstrator system's usability. Eight people played the role of an observer using different communication devices, while a crisis center was performed by a simulator. The simulation also modeled the dynamic nature of disasters. The tasks were created based on a scenario that was fed into the simulator. It used images of real crisis situations. Based on generated events in the scenario, these images were sent to the participants based on their location in the world see examples in Figure 14(b)). The participants were asked to report their observations to the crisis center. Figure 14(a) shows examples of the generated events. All activities were recorded and logged to be analyzed afterward.

All participants accomplished their tasks with relevant observation messages. Preliminary results show that the system is able to integrate the observations and generate appropriate multimodal responses based on available modalities and user contexts. The users often checked the resulted scenarios and rearranged their messages if necessary. Some issues due to unrealistic experiment setup: out of domain reports, unobserved key events and undetected user emotions, occurred and still needed to be resolved.

CONCLUSION

This paper presents approaches for integrating multi-user observations and producing (user-tailored) multimodal information presentation. They can be applied in a multimodal HCI system built based on our proposed framework that is used as a supporting system for crisis management.

Our approach in integrating multi-user messages involves the mechanism of building coherent semantic structures from dynamic concepts. Although a collection of such concepts may be ambiguous and fragmented, it is, by its very nature contextually and temporally correlated. These characteristics, together with the advantageous multimodal redundancy may be used for coming up with coherent, context dependent interpretation of the communicated situation. With the availability of the knowledge of user, of the world and task, such a system can generate integrated and dynamic multimodal responses to the HCI. It is able to realize a coherence and synchronized multimodal presentation with constraints defined by communication technology and context information.

The current approaches do not address the full complexity of any crisis management setting. In our view such controls should be still in human hands, which could mean another HCI via the developed system. This could be done since obtained data still can be verified and updates by multiple observers and intelligent processes. Our proposal includes direct feedback to user inputs, allowing for verifying and altering information and ways for collaborating information. Moreover, the use of ontology, scripts and AIML make inferences easier to verify afterwards. All knowledge may be executed in a different order, but they can be designed and specified sequentially in order of the basic physical events of a crisis scenario. As a drawback, for the system to be fully expressive, all possible scenarios have to be converted into sets of concepts, scripts and AIML in advance since reports about scenarios that have not been specified are not possible. A topic for further research will be deeper evaluations of the integrated system performance, which focus on how people use the system and in particular to assess whether they improve the efficiency of a given task using provided information.

ACKNOWLEDGMENTS

The research is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

REFERENCES

1. Alexandersson J. and Becker T. (2003) The Formal Foundations Underlying Overlay. *IWCS-5*, the Netherlands, 22-36.
2. Carpenter B. (1992) *The Logic of Typed Feature Structures.* Cambridge University Press. England.
3. Cassell J., Stocky T., Bickmore T., Gao Y., Nakano Y., Ryokai K., Tversky D., Vaucelle C. and Vilhjálmsson H. (2002) MACK: Media Lab Autonomous Conversational Kiosk. *Imagina '02*. Monte Carlo.
4. Dor R. (2005) *The Ear's Mind, a Computer Model of the Fundamental Mechanisms of the Perception of Sound*, Master thesis, TUDELFT.
5. Fitrianie S., Poppe R., Bui T.H., Chitu A.G., Datcu D., Dor R., Hofs D.H.W., Wiggers P., Willems D.J.M., Poel M., Rothkrantz L.J.M., Vuurpijl L.G. and Zwiers J. (2007) A Multimodal Human-Computer Interaction Framework for Research into Crisis Management, *ISCRAM 2007*, 149-158.
6. Fitrianie S., Yang Z., and Rothkrantz L. J. M. (2008) Developing concept-based user interface using icons for reporting observations. *ISCRAM 2008*, USA.

7. Foster M. E., White M., Setzer A. and Catizone R. (2005) Multimodal generation in the COMIC dialogue, *ACL 2005*, USA, 45-48.
8. Johnston M, Bangalore S., Vasireddy G., Stent A., Ehlen P., Walker M., Whittaker S. and Maloor P. (2002) MATCH: An Architecture for Multimodal Dialogue System, *ACL'02*, USA, 376-383.
9. Johansson B., Fox A. and Winograd T. (2002) The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms, *IEEE Pervasive Computing*. 1 (2): 67-74
10. Joshi A.K. and Vijay-Shanker K. (1999) Compositional Semantics for Lexicalized Tree-Adjoining Grammars, *Proc. of Computational Semantics*, the Netherlands.
11. Kjeldskov J. and Kolbe N. (2002) Interaction Design for Handheld Computers. *APCHI'02*, China.
12. Mehrotra S., Butts C., Kalashnikov D., Venkatasubramanian N., Altintas K., Hariharan P., Lee H., Ma Y., Myers A., Wickramasuriya J., Eguchi R., and Huyck C. (2004) Camas: a Citizen Awareness System for Crisis Mitigation. *ACM SIGMOD'04*, 955-956, USA.
13. Moore L.K., *CRS Report for Congress: Public Safety Communication Policy*, Confessional Research Service, the Library of Congress, 2006.
14. Norman D. (1993) *Things That Make Us Smart*. Addison-Wesley Publishing Co.
15. Otten J., van Heijningen B., and Lafortune J.F. (2004) The Virtual Crisis Management center - An ICT implementation to canalize information!, *ISCRAM'04*, Belgium.
16. Oviatt S., Coulston R. and Lunsford R. (2004) When Do We Interact Multimodally?: Cognitive Load and Multimodal Communication Patterns. *ICMI '04*. ACM, 129-136.
17. Perlovsky L.I. (1999) Emotions, Learning and Control. *Proc. of International Symposium: Intelligent Control, Intelligent Systems and Semiotics*, 131-137.
18. Pfleger N., Alexandersson J. and Becker T. (2002) Scoring Functions for Overlay and their Application in Discourse Processing. *KONVENS 2002*, Germany, 139-146.
19. Schank R. and Abelson R. (1977) *Scripts, Plans, Goals and Understanding*. Hillsdale, NJ: Erlbaum.
20. Sharma R., Yeasin M., Krahnstoever N., Rauschert I., Cai G., Brewer I., MacEachren A.M., Sengupta K. (2003) Speech-Gesture Driven Multimodal Interfaces for Crisis Management, *IEEE*, 91(9): 1327-1354.
21. Steedman M. and Baldridge J. (2005) Combinatory Categorial Grammar, in: R. Borsley and K. Borjars (eds.) *Non-Transformational Syntax*, Blackwell.
22. Tatomir B. and Rothkrantz L.J.M. (2006) Intelligent System for Exploring Dynamic Crisis Environments, *ISCRAM'06*. B. Van de Walle and M. Turoff, eds., USA.
23. Wahlster W. (2006) Dialogue Systems Go Multimodal: The Smartkom experience. *SmartKom: Foundations of Multimodal Dialogue Systems*. Springer, Heidelberg.
24. Wallace R. S. (2003) *The Elements of AIML Style,* http://alicebot.org.