Information extraction in emergency management missions: an adaptive multi-agent approach

Ana C. Calderon*

Cardiff School of Management,
Department of Computing and Information Systems,
Cardiff Metropolitan University,
CF5 2YB, Cardiff, UK
Email: acalderon@cardiffmet.ac.uk
*Corresponding author

Peter Johnson

Department of Computer Science,
University of Bath,
BA2 7AY, Bath, UK
Email: p.johnson@bath.ac.uk

Abstract: With increasing demands for autonomous agents to work alongside humans in emergency management response (EMR), considerations of translations of human to machine language (and the converse) are timely. We present a prototype where the translation is dealt with by restricting communications to occur through a form of controlled natural language (CNL) (Fuchs and Schwitter, 1995). The prototype is new in that it allows for communications between both physical and virtual autonomous agents, agents are assigned different levels of autonomy, and it includes a level of information hiding that allows for information to be passed to relevant agents, whilst keeping those (humans) involved anonymous. A real-life mission is then used to exemplify how information is retrieved and communicated in the prototype. Finally, some usability experimental results are presented.

Keywords: human and autonomous system interaction; EMR; emergency management response; emergency information system.

Reference to this paper should be made as follows: Calderon, A.C. and Johnson, P. (2017) 'Information extraction in emergency management missions: an adaptive multi-agent approach', *Int. J. Emergency Management*, Vol. 13, No. 3, pp.216–234.

Biographical notes: Ana C. Calderon is a Lecturer in the Department of Computing and Information Systems, Cardiff Metropolitan University and Editor-in-Chief of the CSM's working-papers journal, "Advances in Management and Informatics". Her research interests are in providing formal mechanisms for analysing socially complex situation, such as those encountered in emergency response. Specifically, her work has focused in formalism of individual and collective intent. She was previously a Research Associate in the HASCC (human autonomous systems collective capability) project at the University of Bath, and before that worked on developing software, on a large interactive surface, to support those suffering from dementia. Her PhD was an in-depth analysis of composition in game semantics.

Peter Johnson is Professor of Computing Science at University of Bath. He previously held the posts of Professor of Human Computer Interaction and Head of Department of Computer Science at Queen Mary University of London and Head of Department of Computer Science at University of Bath. He has led EPSRC, EU and Industry research grants continuously from 1986 to the present day. He has over 200 publications in leading computing and engineering journals, conferences and books. He has organised and chaired major international conferences for computing and Engineering societies. He was joint founder and vice-president of the British Computer Society Human Computer Interaction Group. He has advised research councils in UK, Canada and Australia.

1 Introduction

Increasingly we witness the engagement of autonomous agents alongside human agents in emergency, disaster and other complex situations. This offers both advantages and difficulties. The autonomous systems could encompass a varied range of physical agents, ranging from autonomous planes to ground vehicles, and also including virtual intelligent agents able to assist the decision making. The potential advantages include the increased resources available, the increased information gathering and the reduction in risk to the emergency service workers in inhospitable or dangerous situations. The disadvantages include an increased command and control responsibility, a reduction of awareness and trust, an increased information- and work-overload and an increased dependency on rule-based behaviours in situations where the uncertainty and complexity makes it difficult to know what is needed and hence requiring behaviours outside the pre-defined rule-set. The nature of agents involved in emergency management response (EMR) missions vary, some are software agents that contribute to human tasks and carry out various tasks within other socio-technical systems. Others are combinations of software and hardware that control, operate and otherwise engage the combined software hardware system to carry out a task. Some have kinetic and physical capabilities; others have cognitive and emotional ability. Consequently, the interactions with and between, both human and autonomous agents is of concern in such situations. Our research is concerned with addressing these issues of interaction with and between human and autonomous agents in complex and unknown situations, specifically those concerning EMR missions. Our focus is on communications, of two types human-human and human-autonomous agent (physical or virtual). A communications model is presented together with an implementation in a prototypical intelligent interface, highlighting how the system deals with varying autonomy levels and varying degrees of information visibility. The prototype is intended to serve those actually undertaking missions during EMR as well as civilians affected by crisis needing to make complex decisions. The model is written in a form of controlled natural language (CNL) (explained in relevant literature below), making the intelligence in the interaction implementable, adding sophistication to the manner in which humans and machines communicate in order to share and query information and to create new information.

1.1 Controlled natural languages

A CNL is a restricted version of a natural languages (most commonly English), the reasoning behind the restriction is varied, but our interest is in those languages targeted at aiding in the specification of programs (Fuchs and Schwitter, 1995) and in bridging the gap between human language and language that can be processed by machines. For a technical introduction to CNLs we refer the reader to Wyner et al. (2010).

In accordance with similar work, for instance (Xue et al., 2012), our choice was to use ITA CE (Mott, 2010), a CNL consistent with first order predicate logic and based on common logic controlled English (Sowa, 2000).

We make two modifications to ITA CE, needed to deal with composition of information and with varying degree of anonymity. The motivation for anonymity comes from the nature of the situations investigated, namely EMR missions, in which some stakeholders might need to remain anonymous with respect to specific activities, as mandated by organisational doctrines. In addition, missions unfold chronologically and each time new information is added, this changes previously known information and that is what we mean by composition, i.e., how different aspects of incoming are placed together as the mission progresses with time.

There are several branches of research dedicated to minimising the gap between human language and machine language, and CNLs are one example of this. For instance, (Nicola et al., 2014) creates a Kernel language, called SCEL that supports context-awareness, self-awareness and adaptation by representing behaviour, knowledge and aggregation (restricted to specific policies).

1.2 Command and control

Understanding how command and control is affected by crisis is crucial to the success of missions, and we encompass commanding and reporting in our communications model and prototype. Recent directions on formalisation of command have started to move towards command by intent (Moffat, 2011).

Pigeau and McCann (2006) distinguish between commander's intent, common intent and command intent. Commander's intent is the intent of the particular agent responsible for commanding a given mission. Common intent is an idealised concept of all elements in a collective sharing the same intent, whereas command intent is the realistic adaptation to this, that in some specific parts of a mission everyone will share a common intent. The authors of this paper can gladly serve as an example; we each have our own personal goals in life, but with regards to collaborating on this paper we share a common intent and common goals: for it to be published (common intent); to write it unambiguously (common goal), to submit it by a certain date (common goal). Intent and the ability for a common intent to be held is affected by many factors including cultural, personal, organisational and doctrinal views.

Some examples on the formalisation of command includes (Kalloniatis and Fairbairn, 2008) applies the theory of self-synchronisation to command and control. His interest is in modelling interaction between different command approaches which at first glance seems to eliminate the details of the individual command. However his models do allow for the distinction of time scales and interactions between individual processes. The basic idea behind such mathematical models of self-synchronisation is that self-synchronisation is a consequence of the interactions between the elements in the system. The idea is that linking individual nodes which themselves undergo cyclic behaviour causes an emergent collective cyclic behaviour. The model shows how incongruous states evolve to synchronous ones. However, it also copes with the notion of partial synchrony: some elements in the system form synchronous clusters while remaining behave randomly with respect to these clusters. Simulations, which show these clusters, could be used to explain some seemingly synchronous collective behaviour amongst general chaos as can be observed in some real life cases (Calderon et al., 2013). Command and control naturally happens in cycles, e.g., the (Boyd, 1996) observe-orient-decide-act (OODA) loop. Two points to this, the winning team will outpace the enemy's OODA loop and, all decision cycles must synchronise with respect to time and so that decisions interdependent on each other do not occur out of synchrony. Brehmer (2006) expands on this by introducing the concept of a dynamic OODA (DOODA) loop, this model incorporates sense-making, planning and information gathering.

The idea behind a process view is to model command processes as sequences of inputs and outputs and this is concerned with how outputs of one process become the inputs of the next, examples of process views of command are given by HEAT and OODA (which is further modelled in Kallionatis work). This is contrast to the value view of command

Alberts and Hayes (2006), here one is interested in quality: quality of command, quality of intent, quality of information passing, and the authors give descriptions on how to measure these and what their limitations are, for example, the quality of intent might be limited by its expression and its degree of acceptance.

In encompassing commands in our communications model, we are influenced by all the elements from the work just described. Hence we build upon the development of battle management language (Schade and Hieb, 2006) and of a command and control grammar (Hieb and Schade, 2007) (entitled C2LG), where the aim is to give a formal and unambiguous language which can facilitate military communication amongst human and machine agents.

Moreover, understanding command and control via decentralised approaches is also being recognised in settings other than military command and control. For instance (Kota et al., 2012) gives a decentralised method for adaptation in multi-agent systems where self-organisation is possible.

2 Communications model

We will now define a collection of requirements for the construction of an emergency response information system and highlight the desired properties. Our model is specified in language sufficiently formal to make the implementation more faithful to the real world requirements. A formal approach makes such features easier to define, analyse and quantify.

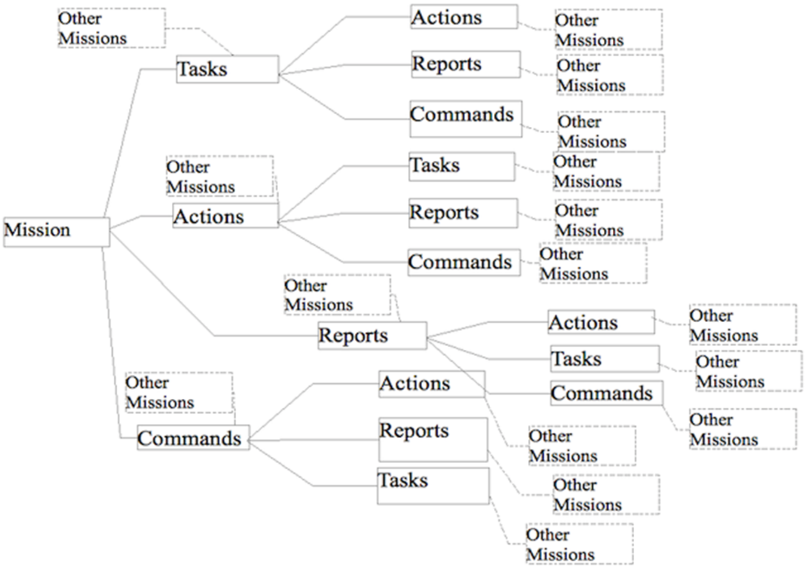
To the best of the authors' knowledge, this is the first communications model with the goal of information extraction in emergency response missions that encompasses autonomous systems, with varying levels of autonomy (further explanation to this is given in the section 'Autonomous Systems of varying autonomy levels').

In keeping with similar research efforts (Eisenstein et al., 2001) we constructed an abstract description of the interactions prior to development. The model, which underlies the prototype, was constructed specifically to suit two particular scenarios of EMR missions (Calderon et al., 2013, 2014). The model is composed of an information structure, dialogue classes and encompasses a notion of information hiding and various autonomy levels.

Our information structure consists of goals, tasks, commands, reports, actions and missions, with the constraint that a mission must be specified prior to everything else, but this is the only hierarchical imposition. One can then decide to interleave these information classes, for instance, to have tasks within actions within commands, and within those commands there may be other tasks, etc. In practice the information structure will unfold depending on what actually happens in an EMR mission and our model does not interfere with the real events sequence, it merely records it as a *placement* in a mission. The reason for requesting that a mission be created first is that having a 'starting point' makes information passing and tracing manageable; in addition, different missions can also be linked. The particular choice of terminology and information structure is a result from case study findings, we are not imposing an information structure, rather we are creating the model to fit structures that seem naturally present in the specific circumstances (field EMR mission of our particular case studies). Our model advances understanding past case studies, formalising (software) requirements for information retrieval, passing and communications in the particular observations, followed by an implementation.

The information structure is represented in Figure 1, showing the flexibility in the definitions of components and also which components are atomic to others. For example, to define a mission, one must define a task, an action, and a command, if it is related to other existing missions, this relationship must also be made known (this, in practise, is done automatically in our prototype). Further, to specify a command, one must specify which actions, reports and tasks are associated to that command. The individual components are formally defined (in ITA CE) below, but the definitions are easier to understand using Figure 1 as a visual representation of the relationship between the components of the model.

Figure 1 Information structure underlying prototype



However, before the definitions can be presented, we need to modify ITA CE by adding one definition, the ability to compose both sequentially and in parallel. Rather than using mathematical symbols for these, as is common practice, we felt it would be more compatible with CE to write ‘AND’ for things that are created in parallel and ‘THEN’ for things that are created sequentially. Note that this is not an ‘if then statement’ (which is already covered by CE). Another modification is that in ITA CE once ‘agent known as *X*’ is specified there is no assumption that all subsequent mentions of *X* refer to the same agent. We however give a unique identifier to each agent, this is translated into the prototype by simply requiring that agents be registered in our system and that they log in each time they use it.

Our system also has a form for individual and personal communications that are not intended to be shared with other agents, that is intended to be kept as part of a ‘personal diary’ of a particular agent i.e., that can be linked to missions, tasks, activities, commands and reports. Even though the personal space of the system allows for people to write in CE or in plain English, they are required to provide some specific information to aid in the translation from plain English to CE, should they choose to share that information at a later date, and also to categorise their information posting in accordance with our model.

Goals and activities are essential building blocks for our model. An activity is simply an ‘active state’, an agent is undertaking some activity if it is ‘doing something’. Activities are an integral part of tasks and missions. A goal is essentially something an individual or a group of agents wishes to accomplish or something they desire to become true within a given task and/or mission. Goals are an integral part of tasks and missions and can be permanent (for the lifetime of a particular task and/or mission) or temporary. The specification for a goal is as follows:

Conceptualise a ~ goal ~ G that
has the timeframe T as ~ timeframe ~ and
has the desired result D as ~ desiredResult and
has agents A_1, …, A_n *as ~ members.*
Conceptualise a ~ permanent goal ~ G that
is a goal and has the permanent stamp P as ~ stamp ~.
Conceptualise a ~ temporary goal ~ G that
is a goal and has the temporary stamp K as ~ stamp ~.

A task is specified as:

Conceptualise a ~ task ~ T that
has the goal G as ~ goal ~ and
has the subtasks Ts_1, …, Ts_n *as ~ subtasks ~ and*
has the activities A_1, …, A_n *as ~ activities ~ and*
has status St as ~ status.
Conceptualise the ~ task ~ T that
~ is visible to ~ the agents A_1, …, A_n *and*
~ is from ~ the locations L_1, …, L_n.

It is thus required that when posting a new task, agents specify a task-related goal, subtask (or lack thereof), task activities and who is allowed to see full details of the task. Note that geographical locations are automatically added, although these can be manually input (in the prototype this is via the button on the bottom right). The status of a task is altered in the system once tasks are considered completed.

Commands are defined in a 'command by intent' manner (Moffat, 2011). The specification of a command is obliged to detail an *intent*, and if one wishes to define a *method* together with that intent, then one must specify an M-command (see definition below), a command does not have a method as a requirement, but it must specify a list of expected house rules which are ways, agreed by stakeholders in a particular mission, that certain commands are typically followed (if no such rules exist then 'none' is written in their place). If a new way of completing a command proves to be successful (without further requirements that might measure success against the standard expected house rule) then it is added to the list of house rules. Commands (and reports) often require a response, and this is achieved in our model via the definition of a *RC-command*. The full definition of commands is given bellow:

> *Conceptualise an ExpectedHouseRule ~ is StandardWayCompletion or is NewWayNonFailure.*

ExpectedHouseRules (specified above) is essentially a collection of protocols and rules that specify ways in which a particular collective (or organisation, or coalition) expects others to respond to particular commands. This list is intended to grow, hence if an agent creatively acts in a novel way, that will be added to the collection provided it was deemed successful. If an expected house rule already exists and a new way is discovered then it will be added if, in addition to being successful, it is not worse (for some measure of success) than the current rule; but it does not need to be 'better' than the *ExpectedHouseRules* way of completing that command.

> *Conceptualise a command (C, status: St) that*
> *~ has timeframe ~ T and*
> *~ has startstate ~ S and*
> *~ has endstate ~ E and*
> *~ has intent ~ I and*
> *~ has expectedhouserules ~* $Z_1, \ldots, Z_n$ *~ and*
> *has status St as ~ status.*
>
> *Conceptualise the command C*
> *~ is from ~ the agent A and*
> *~ is to ~ the agents* $A_1, \ldots, A_n$ *and*
> *~ is visible to ~ the agents* $A_1, \ldots, A_n$ *and*
> *~ is from ~ the locations* $L_1, \ldots, L_n$ *and*
> *(~ is from ~ the command (C, status: St) or ~ is from ~ individual information I).*

Conceptualise a RC-command ~ (RC-C, status: St) that is a command and has required response ~ RC.

Conceptualise an M-command ~ (C, status: St) that is a command and has method ~ M.

Note that commands are specified with a ‘status’ attached to them, a status can be either ‘open’ or ‘closed’ and that is representative of whether the goal of the command has been accepted as achieved. A report is a means to convey information, and so does not set goals, hence there is no need for a status, the specification is given below:

Conceptualise a report R that

~ has timestamp ~ T and

~ has details ~ D ~ and

has status St as ~ status.

Conceptualise the report R

~ is from ~ the agent A and

~ is to ~ agents $A_1, \ldots, A_n$

~ is visible to ~ the agents $A_1, \ldots, A_n$ *and*

~ is from ~ the location L$.

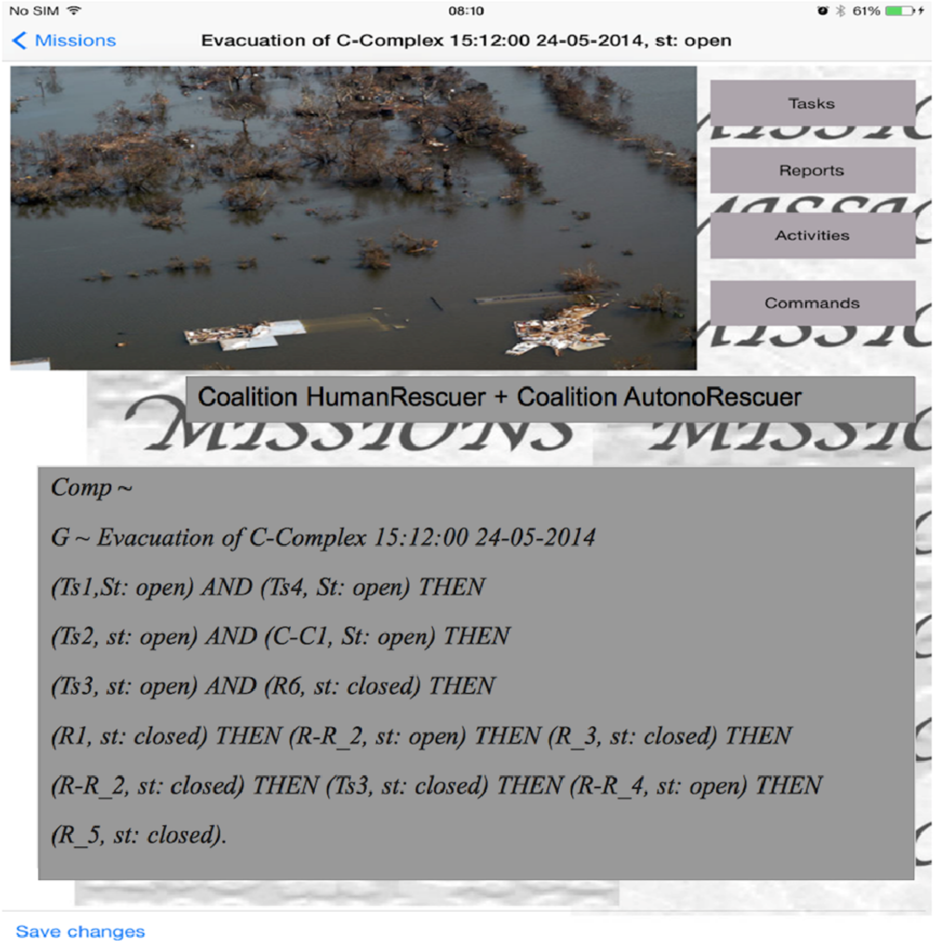
We also define a notion of a report that allows for a reply to be requested, in this type of a report a status has the meaning of whether the request had been given and accepted. The specification is given bellow:

Conceptualise a ~ Req-report ~ (Req-R, status: St) that is a report and ~ has required response Req.

Across our information structure, there is a notion of visibility (detailed in the ‘Anonymity and information hiding’ section); some agents are able to see certain missions, tasks, commands, reports, activities. This is not hereditary, for instance if agent X has visibility to command C, and command C contains report R, then it is not necessarily true that X can see details of report R. The description of tasks follows a similar style to commands and reports, so they must specify the agent creating the particular task or activity and who is allowed to see it, their location and current status. In addition, tasks are composed of goals, subtasks, activities.

Missions are a bit more complex and we detail them now. In what follows we will use ‘&’ as a placeholder symbol for AND (parallel composition) or THEN (sequential composition). Missions are dynamic and encompass all elements already defined, we specify a ‘placement’ of all aspects of a mission and explain how they have composed at the end of a mission (for a mission with closed status) or are currently composed (for a mission with open status); a placement tells the order in which different aspects have occurred or are occurring (depending on status) and implicit in the information will also be their structure. A mission will have goals, tasks, activities, commands, reports, status and a placement dependent on the current status of the mission. An example of a mission in a real EMR mission can be seen in Figure 2, while Figure 3 shows the specification for the reports of that particular mission (users can scroll the report area to view their history). In addition commands, reports, actions and tasks within a mission have a ‘status’ association with the particular time in the missions; a status can be ‘open’ or “closed”. Commands and reports that request a response (RC-Commands and Req-Reports) are set to “open” by default, and those that do not are set to ‘closed’ by default. And, once a request is accepted the status of a RC-Command or a Req-Report is changed to closed.

Figure 2 An evacuation mission, unfolding events can be read by autonomous agents (see online version for colours)



Evacuation of C-Complex 15:12:00 24-05-2014, st: open

Coalition HumanRescuer + Coalition AutonoRescuer

Comp ~

G ~ Evacuation of C-Complex 15:12:00 24-05-2014

(Ts1,St: open) AND (Ts4, St: open) THEN

(Ts2, st: open) AND (C-C1, St: open) THEN

(Ts3, st: open) AND (R6, st: closed) THEN

(R1, st: closed) THEN (R-R_2, st: open) THEN (R_3, st: closed) THEN

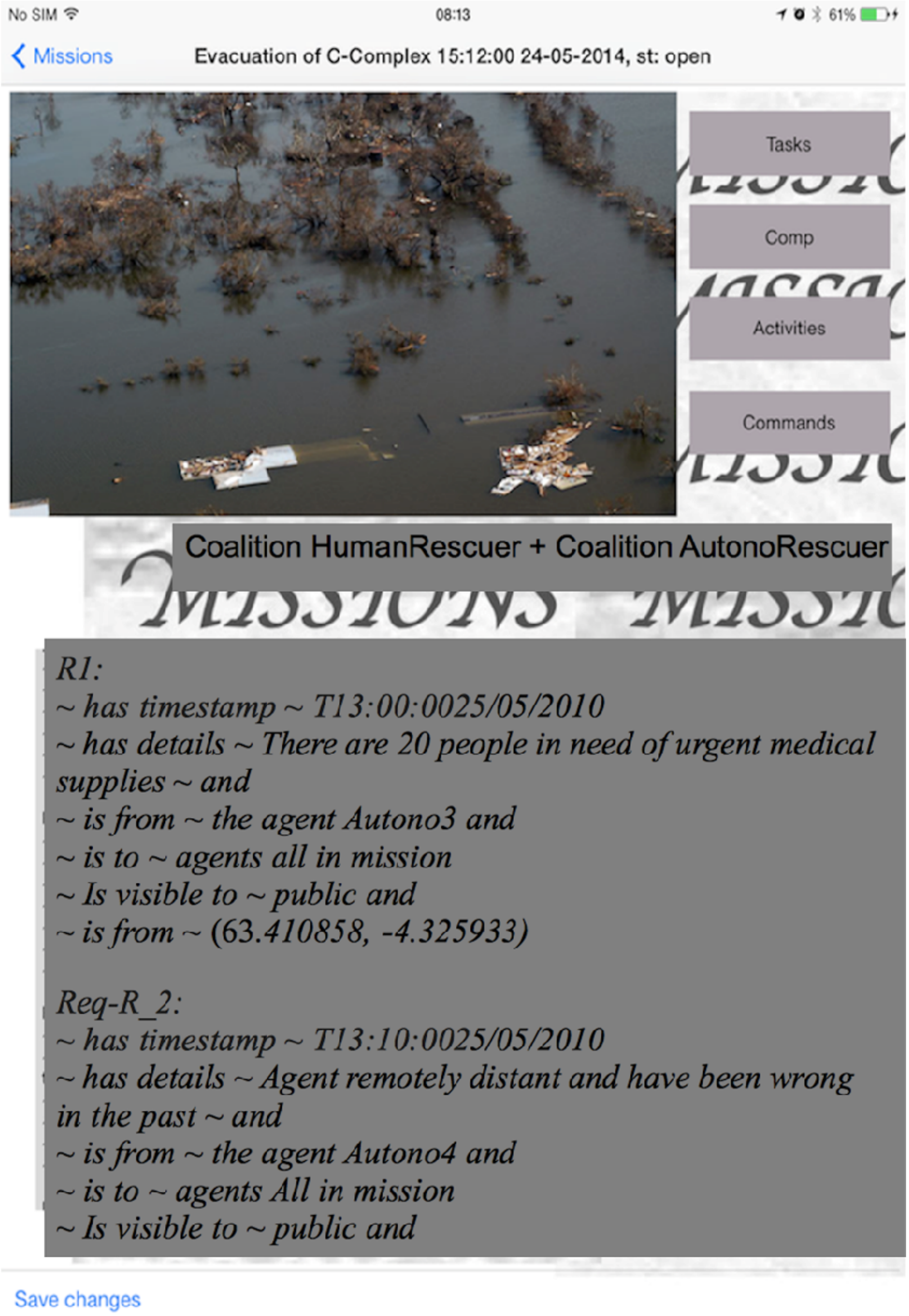
(R-R_2, st: closed) THEN (Ts3, st: closed) THEN (R-R_4, st: open) THEN

(R_5, st: closed).

An agent wishing to view the full specification of all tasks, commands, reports or activities within the mission (up to present time), can achieve this by selecting one of the grey buttons on the right. Figure 2 shows the specification for two of the reports in the evacuation mission of Figure 1.

The status of a report, command, task and activity or mission is altered automatically in the system once tasks are considered completed. The system will also link these information classes according to their creation, for instance if a command is created from a task page, these will be linked with the command being assumed a ‘sub-category’ of that task.

Figure 3 Reports within a particular mission (see online version for colours)



Evacuation of C-Complex 15:12:00 24-05-2014, st: open

Coalition HumanRescuer + Coalition AutonoRescuer

R1:
~ has timestamp ~ T13:00:0025/05/2010
~ has details ~ There are 20 people in need of urgent medical supplies ~ and
~ is from ~ the agent Autono3 and
~ is to ~ agents all in mission
~ Is visible to ~ public and
~ is from ~ (63.410858, -4.325933)

Req-R_2:
~ has timestamp ~ T13:10:0025/05/2010
~ has details ~ Agent remotely distant and have been wrong in the past ~ and
~ is from ~ the agent Autono4 and
~ is to ~ agents All in mission
~ Is visible to ~ public and

It is worth noting that both Figures 2 and 3 are specifications, in our prototype, of a mission from a real-life case study during Hurricane Katrina (specified in Calderon et al., 2013).

The dialogues we are interested in all concern acquisition and distribution of information in a manner that aims at easing the cognitive load of humans involved. The goal is to incorporate autonomous agents (virtual and physical) with the purpose of aiding the humans responsible for the decision-making. To this end we categorise human – autonomous agent communications into three *dialogue classes*:

- *Asking for new information*

This is achieved through a Req-Report, an RC-Command or via setting up a new task with an information goal.

- *Setting an overall purpose*

This does not require a response and is achieved through a command or through a task.

- *Assigning activities and tasks*

This is achieved through a command or through a task (a task can be used to assign subtasks).

2.1 Anonymity and information hiding

It is often the case that an agent knows information relevant to another agent but both are unaware of it, or the agent might wish to share the information whilst hiding some details of it, we now explain how our system deals with it. To that end, we define a named agent in a mission A created by agent Y, to be an agent X such that Y has specified that X can see all details of that particular mission. Now, consider a mission A created by agent Y where agent X and agent Z are amongst named agents (so have full visibility to it), then the following conventions are part of our communication model and thus implemented in our prototype:

'Naming' is not hereditary, so if agent Z creates a task (or report, command, etc.) in mission A above, agent X will not have visibility to it, unless agent Z names agent X in the task, regardless of the fact that agent X has visibility to the 'parent mission'.

Agents X and Z are allowed to edit mission A but they can only add information to it, and they cannot name other agents. The reason for this imposition is that mission A was created by agent Y, and so the named agents on it must respect the conditions imposed by agent Y. If they feel an agent should be added or that information should be deleted, they must either comment on that mission to that effect, or make a report to agent Y requesting that more agents be added to it.

Agent Y is allowed to remove information from it, but that removal will be recorded as an event and agent Y cannot remove agents post mission created, he can only name more agents, not less. If agent Y wishes to add information not visible to all agents, he must narrow that visibility by using the non-hereditary property of visibility, so he must create a task, report, added information, etc. and name fewer agents.

Awareness must be drawn to the fact that named agents need not be engaged in the mission, task, etc. that they are named in. Whenever an agent changes her status, creates a new action, etc. this will appear to all users of our system as an 'event', named agents on that mission, etc. will be allowed to see more details, including location details.

The idea behind having some general events visible to all users is in case a user feels more information on a particular event would be especially important to something he is currently doing, then he can submit a report requesting a higher level of visibility (a request is a special kind of report where a response is required).

2.2 Autonomous systems of varying autonomy levels

To incorporate varying autonomy levels compatible with similar research efforts, we consider the types and levels of autonomy give in Parasuraman (2000). We have considered levels up to 5 on Sheridan's 10 levels scale (Sheridan, 1978) since higher levels are not likely. For the sake of readability we reciprocate the levels of automation relevant to our work. These are:

- humans must do the entire work themselves, the autonomous agents offers no assistance
- the autonomous agent offers a complete set of decision and action alternatives
- in addition to 2, the autonomous system is capable of narrowing the selection to fewer alternatives
- in addition to (1, 2, and 3) the autonomous system is able to selecting just one alternative and offering it.
- in addition to (1, 2, 3 and 4) the autonomous system is capable of executing that suggestion, if permitted by a human.

Moreover, allocation of decision rights is crucial during EMR and it is encompassed into the varying levels of autonomy, but the particular allocation must be left at the discretion of all stakeholders involved in the aftermath of a disaster. Hence, the decision of allocation rights in our model is left at the responsibility of individual agents with two constrictions.

Agent X can permit/forbid decision rights to agent Y if and only if agent Y is of lower autonomy level than agent X, and the permission/prohibition does not contradict organisational or coalition hierarchies.

3 Example

The mission chosen for our example (Figures 2 and 3) centres around an ex-marine who evacuated hundreds of people trapped in a building, in the aftermath of Hurricane Katrina (Calderon et al., 2013). A large portion of those needing evacuation were elderly, disabled, or in need of urgent medical assistance or supplies. His mission was to evacuate all people trapped, we altered it to incorporate autonomous systems and consider how part of that particular case study is described in our communication model. To exemplify our communication structure, we focus on a particular portion of the mission chosen. Namely we focus on communications that occur in order to determine evacuation priority for instance how many of the evacuees are disabled and how many are in urgent need of medical supplies.

Figure 2 shows the composition of the mission, to view the particular details of each task, command or report, one must select the desired category (top right corner). An example with selected reports is given in Figure 3. We will specify the mission by information category before describing the composition.

Both Figures 2 and 3 show how our prototype decomposes elements from the mission and communicates them to agents, in ITA CE.

Tasks

The mission can immediately be divided into three tasks. The first task is to determine number of people trapped in building, which can then be subdivided into determining the number in need of medical supplies and the third task is to determine the number in need of urgent medical supplies. For example, the task (T_3) to determine the number of urgent medical supplies is specified as follows:

T_3:

> *~ has timestamp ~ T13:00:0025/05/2010*
>
> *~ has details ~ There are 20 people in need of urgent medical supplies ~ and*
>
> *~ is from ~ the agent A.S. 3 and*
>
> *~ is to ~ agents all in mission*
>
> *~ Is visible to ~ public and*
>
> *~ is from ~ (63.410858, –4.325933).*

We also consider a fourth task, to determine the structural condition of the building. This task will have a command, asking for autonomous systems to be sent in to determine the extent of water damage in the basement with a report giving a response of whether the building is likely to collapse in the next week.

Reports

Reports should be given to numbers of evacuees as specified in the tasks. For example, a report in response to the number of people in urgent need of medical supplies, from a human agent can be found in Figure 2 (report R_1). The second report (Req-report R_2) in Figure 2 is that of an autonomous system of level 4 contesting the number, with the reason given that the agent has estimated similar numbers wrongly in previous missions. As the mission progresses, another report is sent from an autonomous system of autonomy level 4 to confirm how many evacuees are in urgent need of medical supplies as follows:

R_3:

> *~ has timestamp ~ T13:15:0025/05/2010*
>
> *~ has details ~ I can confirm that there are 20 people in urgent need of medical supplies, will need supplies in the next 24 hours. This is final report regarding* Ts_3 *~ and*
>
> *~ is from ~ the agent Human in building and*
>
> *~ is to ~ agent A.S. 4*
>
> *~ Is visible to ~ public and*
>
> *~ is from ~ the location (51.4006150, 2.466468120).*

The remainder of the communication to establish number of evacuees of varying levels of urgency. These reports are all given through our prototype (in the scrollable area where R_1 and R_2 are given in Figures 1 and 2).

Commands

A command is given to specify which agents must be sent in order to count number of evacuees and determined the structural integrity of the building. For instance, to determine the damage due to basement flooding the command is given as follows:

RC-*C*1:

~ has timeframe ~ 300minutes and

~ has startstate ~ current state and

~ has endstate ~ current state (no changes) and

~ has intent ~ send in Autono3 to decide extend of water damage in basement requesting a response of whether building likely to collapse in the next week ~ and

~ has expectedhouserules ~ a.s.3 will report to a.s.4 to make decision ~ and

~ is from ~ the agent A.S.4a and

~ is to ~ the agents all of a.s. Levels 3 and up and humans and

~ is visible to ~ public and

~ is from ~ the locations (51.378014, –2.32593), (63.410858, –4.325933), (51.4006150, 2.466468120) and

has required response ~ explanation of water damage.

Composition

Part of the mission composition can be seen in Figure 1, reciprocated below with natural language explanations:

G: Evacuation of RC-Complex 15:12:00 24-05-2014

The goal, the time and date it was set.

(Ts_1*,St: open) AND (*$\$Ts_4$*, St: open) THEN*

Tasks to determine number of people trapped in building (with subtasks: Ts_2 THEN Ts_2) and to determine the structural integrity of the building. These tasks can happen in parallel, and are then followed by

(Ts_2*, st: open) AND (RC-C1, St: open) THEN*

subtask to determine number in need of medical supplies and a command to send in agents who are human, or of autonomy level at least 3, to decide extend of water damage in basement with a mandatorily required response of whether building likely to collapse in the next week. This is then followed by

(R_7*, st:closed) AND (*Ts_3*, st: open) THEN*

a report by an autonomous agent of level 2 specifying that the moisture level of the air is 80%. The status of this report is ‘closed’ as it does not expect a response. This happens simultaneously with a task to determine number in need of urgent medical supplies is then opened. Two other reports then follow

(R_8*, st:closed) AND (*R_9*, st:closed)*

that use report R_7 to estimate that the moisture level has risen by 5% in the past 24 h and that the wall structure intact.

(R_6, st: closed) THEN

These three most recent reports can then be used to estimate that the damage is not severe enough to cause alarm with regards to building collapsing. As the mission progresses more reports, responding to previously set tasks arise, and take form of conversations as follows:

(R_1, st: closed) THEN

Agent of autonomy level 3 declares that there are there are 20 people in need of urgent medical supplies, then

(Req- R_2, st: open) THEN

Agent of autonomy level 4 declares that the particular agent is remotely distant and has been wrong in the past, requesting a confirmation response from all agents involved in the mission.

Our prototype also contains a space dedicated to face-to-face interaction (focused for human users only). The interactive pages contain tools for writing and sketching, to aid in visual explanations of missions, commands, or reports.

Another portion of our prototype, that is also solely dedicated to humans is the ability to log on with varying identities. This is best motivated by example, consider, for instance ‘husband’, ‘scientist working for organisation X’, ‘grandfather’; these can all be used to describe the same person who will act differently in an interaction on the same topic, depending on which aspect of his identity is relevant at the time. The same applies during EMR and our system is designed to respect agents” varying identities, the system encompasses a notion of *identity* by allowing users post information as themselves, as members of an organisation they work/volunteer for, or as members of a coalition (or of an organisation belonging to a coalition).

4 Prototype evaluation

Three main aspects of our approach were tested: the communication structure through the prototype, the general design of the prototype and we also observed communications in a controlled environment mimicking conditions expected during EMR missions. Before we present the experimental results we will highlight some further particulars of our prototype available to humans, not covered by the model.

In addition to what we have described in our communications model, the prototype allows humans to search for information within any of the information classes, and narrow it by locality, creation date or linkage to other information. It also allows for pictures and sound files to be attached to any new posting.

Moreover, it contains a space dedicated to face-to-face interaction, where humans can write and sketch whilst engaging in conversations during activities, for instance if a commander needs to explain to others further details of a command, this was inspired by what is observed in real-life scenario. Humans can also keep some information added to any information class, this can be written in plain English, and is only visible to who created, this is a personal space dedicated area. Finally, as social beings, humans have several identities or personas intertwined in our daily activities. For instance husband, scientist working for organisation X, grandfather can all be used to describe the same person who will act differently in an interaction on the same topic, depending on which aspect of his identity is relevant at the time. This is no different during EMR and our system is designed to respect agents varying identities, the system encompasses a notion of identity by allowing users post information as themselves, as members of an organisation they work/volunteer for, or as members of a coalition (or of an organisation belonging to a coalition).

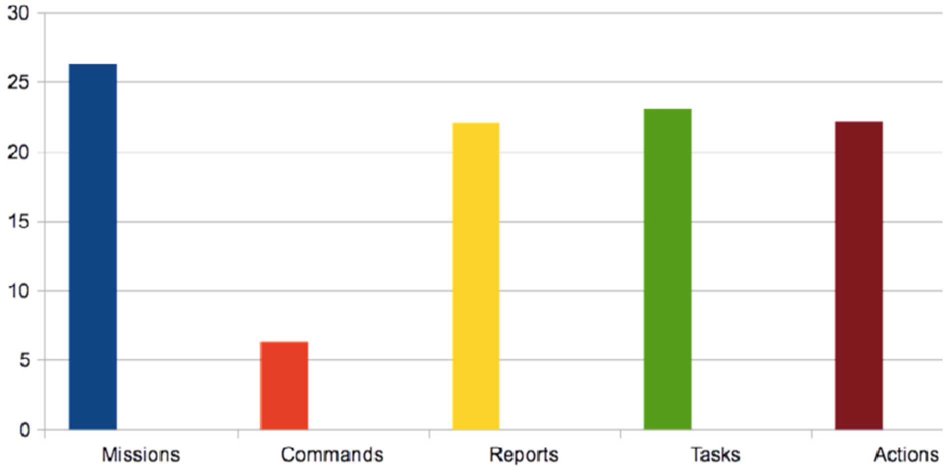
4.1 Evaluation of communication model with regards to interface

To evaluate the information structure (division into missions, tasks, reports and commands) is sensibly handled by our system, we conducted a longitudinal experiment of usability (Karapanos and Hassenzahl, 2012).

To that end, we uploaded the prototype on mobile the devises of the participants and monitored the usage within each information division for the duration of one month. This included 8 participants with an average age of 38.8 years of age. The average percentages of objects created at the end of the experiment can be seen in Figure 4 (in decreasing order, the categories with the highest number of objects were missions, tasks, reports, finally commands).

It would not have been reasonable to expect that all classes of information were used exactly equally, and we attribute the small percentage of commands to the nature of the actual study. The variance in the data was 61.31, which decreases significantly to 3.88 if commands are ignored from the data. We believe the nature of the study itself is the reason for this; participants were asked to use the software as they saw fit, and it is unlikely that they would be comfortable commanding others unless instructed to do so. Our conclusion is that participants were comfortable in creating all the necessary information and dividing it in accordance to the communication categories through our system.

Figure 4 Information distribution (see online version for colours)



4.2 Evaluation of human-autonomous systems communications

To test how human-autonomous systems communications are handled by our system, we conducted a Wizard-of –Oz lab-based experiment, involving a total of 12 participants.

Participants were placed in a simulation of a post-disaster operation, and were instructed to focus on decision-making capabilities, they assumed the role of a commanding officer, and had to gather incoming information from other commanding officers, from civilians, and from emergency relief personnel at the scene. The scenario for the simulation was that described in the previous section (see also the screenshots in Figures 2 and 3). Incoming information was sometimes conflicting, and participants used our system to create missions deploying (human or autonomous agent) to the scene to provide specific relief, food, rescue, etc. We were interested in how the participants used the system in a simulated emergency, and also whether there were differences in accepting information from humans vs. autonomous agents. We analysed the proportion, per participant, of ignoring or accepting information from autonomous and human agents. By this we mean whether the participant decided to take the information into consideration or not in his decision-making or if the participant decided the information was false. This was measured by observation and confirmed with post-experiment interviews. The difference in data between autonomous and human had a *t*-test value of approximately 0.949.

5 Conclusion

We have presented an intelligent interface for communications between humans and autonomous agents (and communications amongst humans) during EMR missions. The specifications for the communication requirements were fully written in a CNL, prior to development. The communications structure we defined has four classes of information (commands, reports, activities and tasks) and encompasses three classes of dialogue aimed at dealing with acquisition and querying of information. In addition, to aid in the translation from human to machine language, communications through the prototype are restricted to a form of CNL, thus enabling autonomous agents to communicate with our system. The non-human agents considered are both physical systems and virtual intelligent agents, and we consider autonomous agents of varying autonomy.

In addition to a communication model and prototype, we presented some empirical assessment of the information structure in our model, through the prototype. We also assessed the usability of our interface. The fact that the model was written in controlled language make it implementable, and thus we have presented an intelligent interface for communications between humans and autonomous agents during missions of emergency response. The novelty in our approach is also given by our varying autonomy level and varying levels of information hiding contained in our prototype.

At present, the only hierarchical imposition of our model is that missions must be specified before anything else. Immediate future work requires understanding variations with a hierarchical information structure, whilst keeping the division into information classes (missions, tasks, etc.) constant. For instance, should we allow for tasks, etc. to be created as standalone. And if so how would this impact on our model, which would then cause a reiteration of the prototype. Further future work includes how to incorporate sensing software, see for instance, Fuchs and Schwitter (1995) and Wyner (2010) in our prototype, and understanding what implications these might have for our communication model.

Acknowledgements

Part of the research reported here benefited from funding from EPSRC EP/J012521/1 Human Autonomous Systems Collective Capability (HASCC), we are grateful to the EPSRC for this funding.

The picture shown of the flooding (Figures 2 and 3) is in the public domain, courtesy of the US Geological Survey. The copyright policy can be found in the USGS homepage, http://www.usgs.gov

We would also like to express deep gratitude to the anonymous reviewers and to the journal editors for their helpful comments that greatly improved the quality of this journal.

References

Alberts, D.S. and Hayes, R.E. (2006) *Understanding Command and Control*, DOD Command and Control Research Program, CCRP Publication Series.

Boyd, J.R. (1996) *The Essence of Winning and Losing*, Unpublished Lecture Notes, Retrieved from pogoarchives.org

Brehmer, B. (2006) 'One loop to rule them all', *Proceedings of the 11th International Command and Control Research and Technology Symposium*, Cambridge, UK.

Calderon, A., Hinds, J. and Johnson, P. (2013) 'Leading cats: how to effectively command collectives', *10th International ISCRAM Conference*, Baden-Baden, Germany, pp.32–41.

Calderon, A., Hinds, J. and Johnson, P. (2014) 'IntCris: a tool for enhanced communication and collective decision-making during crises', *11th international ISCRAM Conference*, Pensylvania, pp.205–219.

Eisenstein, J., Vanderdonckt, J. and Puerta, A. (2001) 'Applying model-based techniques to the development of UIs for mobile computers', *Proceedings of the 6th International Conference on Intelligent User Interfaces*, ACM, Santa Fe, NM, USA, pp.69–76.

Fuchs, N.E. and Schwitter, R. (1995) *Specifying Logic Programs in Controlled Natural Language*, arXiv preprint cmp-lg/9507009.

Hieb, M.R. and Schade, U. (2007) *Formalizing Command Intent Through Development of a Command and Control Grammar (i-069).*

Kalloniatis, A. and Fairbairn, D.E.A. (2008) 'New paradigm for dynamical modelling of networked c2 processes', *Proceedings of the 13th International Command and Control Research and Technology Symposium (2008)*, UK, pp.17–19.

Karapanos, E., Jain, J. and Hassenzahl, M. (2012) 'Theories, methods and case studies of longitudinal HCI research', *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, ACM, May, Austin, Texas, pp.2727–2730.

Kota, R., Gibbins, N. and Jennings, N.R. (2012) 'Decentralized approaches for self-adaptation in agent organizations', *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 7, No. 1, p.1.

Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V. and Thomas, S. (2002) 'Relevance models for topic detection and tracking', *Proceedings of the second international conference on Human Language Technology Research*, Morgan Kaufmann Publishers Inc., San Francisco, USA, pp.115–121.

Moffat, J. (2011) *Adapting Modeling and Simulation for Network Enabled Operations*, Tech. Rep., DTIC Document.

Mott, D. (2010) *Summary of Controlled English*, ITACS, https://www. usukitacs.com (Retrieved 21 September, 2016).

Nicola, R.D., Loreti, M., Pugliese, R. and Tiezzi, F. (2014) 'A formal approach to autonomic systems programming: the SCEL language', *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 9, No. 2, p.7.

Parasuraman, T.B. (2000) 'A model for types and levels of human interaction with automation', *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 30, No. 3, pp.286–297.

Pigeau, R. and McCann, C. (2006) *Establishing Common Intent: The Key to Coordinated Military Action*, The Operational Art: Canadian Perspectives-Leadership and Command.

Sayyadi, H. and Raschid, L. (2013) 'A graph analytical approach for topic detection', *ACM Transactionson Internet Technology (TOIT)*, Vol. 13, No. 2, p.4.

Schade, U. and Hieb, M.R. (2006) 'Formalizing battle management language: a grammar for specifying orders', *2006 Spring Simulation Interoperability Workshop*, Huntsville, AL.

Sheridan, T.B. (1978) 'Supervisory control of remote manipulators for undersea applications', *Proceedings: International Conference on Cybernetics and Society, Institute of Electrical and Electronics Engineers*, Tokyo, Japan, p.237.

Sowa, J.F. (2000) 'Ontology, metadata, and semiotics', *Conceptual Structures: Logical, Linguistic, and Computational Issues*, Springer, pp.55–81.

Wyner, A., Angelov, K., Barzdins, G., Damljanovic, D., Davis, B., Fuchs, N., Hoefler, S., Jones, K., Kaljurand, K. and Kuhn, T. (2010) 'On controlled natural languages: properties and prospects', *Controlled Natural Language*, Springer, Berlin Heidelberg, pp.281–289.

Xue, P., Poteet, S., Kao, A., Mott, D., Braines, D., Giammanco, C. and Pham, T. (2012) *Information Extraction Using Controlled English to Support Knowledge-Sharing and Decision-Making*, Tech. Rep., DTIC Document.