A Multi-Agent Context-Management System for RECON Intelligence Analysis

Alexis Morris, William Ross
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada
{alexis.morris, william.ross}@unb.ca

Mihaela Ulieru
School of Information Technology
Carleton University
Ottawa, ON, Canada
mihaela@theimpactinstitute.org

***Abstract*—Adaptive systems require technologies that enable high synchronicity between its users and their unfolding situation dynamics, in concert with system response actions. To be effective, a multi-dimensional view of context must be considered and incorporated so as not to limit its application potential. This work advances the development of RECON, an initiative to support intelligence analysts, with a novel context-management approach. The central concepts involved in the management of explicit and implicit contexts such as situational, system, and user psycho-physiological states are presented, along with the resulting multi-agent system design. In particular, a community of generic, expert service-oriented agents is proposed, supported by a context-sensitive cognitive model (based on COCOM), to facilitate and improve system adaptations. These pave the way toward future developments and experiments in improving human-machine synergy with adaptive context-management systems.**

***Index Terms*—Adaptive systems, context awareness, context-management, multi-agent systems, service-oriented agents**

I. Introduction

Information overload presents a critical challenge to the development and success of advanced socio-technical systems, where humans-in-the-loop must make sense of an ever-increasing inflow of data in order to perform their tasks. Information generation enables unique perspectives of the world to be derived from raw data and facts; however, with increases in data volume, there is a need for better management of how this information is processed, presented, and consumed, especially in time-sensitive situations. With advances in technology, it has become increasingly important for decision-makers and strategists to make sense of world dynamics quickly, in order to stay ahead of unfolding situations. Failure to maintain control of dynamic situations (e.g., due to poor information management) can be the cause of small annoyances like reduced inefficiency, or even larger catastrophes [1]. In both cases, efficient management is vital, particularly as the trend towards a more information-dependent society continues.

To manage the dynamics of real-world information monitoring and sense-making, there are many different explicit and implicit contexts that can be considered by an adaptive system. However, the challenge is in finding the "right" context so that the system, in turn, can act as an aid (rather than a hindrance) to the expert human user. As in [2], *context* is considered as anything that can be used to correctly identify the situation of a user. Explicit context can be provided directly by the user or generated based on the user's actions, such as system tasks recently performed (based on system logs) and current location data (based on mobile global-positioning systems) [3]. Implicit context, on the other hand, provides a less concretized notion, describing users' psycho-physiological states such as their current cognitive mood and stress level. These can be obtained through active and passive sensing of users via bio-metric sensors, although such states can also be deduced from explicit sources such as camera monitoring of facial expressions [4].

The successful management of both kinds of context is important. Systems that are adaptive to the dynamics of a wide range of contexts can increasingly support properties favouring the "5 Rights" [5]—i.e., providing the right information to the right person in the right place, at the right time, and in the right way (e.g., based on the preferences of the user). Such highly adaptive systems are becoming a focus at present [6], and this work aims to contribute a possible path forward in their development and practical usage, especially in the intelligence analysis domain.

A. RECON Intelligence Analysis

The RECON (REcommending Cases based on cONtext) system is a recent initiative aimed at providing a capability for intelligence analysts that takes into account their need for relevant information consumption in a time-sensitive environment. As a part of Defence Research & Development Canada's iVAC (Intelligent Virtual Analyst Capability) project [7], RECON uses an adaptive-systems approach for information offloading and filtering to assist intelligence analysts. Its architecture has been outlined previously in [2] and combines the following components: human-computer interface (HCI), data collector, case-based recommendation (CBR), brain-computer interface (BCI), and context manager. It is envisioned that such a unique combination of components, enhanced through the use of explicit and implicit context management, can better support analysts in performing their tasks by satisfying the different information "rights" mentioned above.

Intelligence analysis provides an interesting domain for the study of how to develop a flexible and adaptive technological solution to the problem of information overload, and the present work focuses on extending the current RECON system

by introducing a multi-agent, context-management architecture. This approach, designed with flexibility and decentralization in mind, has at its core a community of service-oriented expert agents that can be extended to a host of applications and user domains. This paper contributes to the discussion of adaptive context-management systems with a proposed architecture based on human factors and a context-sensitive cognitive model.

The remainder of the paper is organized as follows. Section II provides a human-factors perspective to context management, which is used in both Section III and Section IV, respectively, to present the context-management architecture in detail and to highlight the use of the cognitive modelling paradigm in context management. Section V then discusses related work in the field and highlights the novelty of the proposed context-management approach, while Section VI presents the conclusion and offers direction for future work.

II. Human Factors for Context Management

The "human factor" is considered to be a significant contributor to effective system design and run-time operation [1]. However, it is difficult to incorporate human behaviour into system designs due to the inherent multi-dimensional nature of socio-technical systems [8], [9]. Such systems, as presented in [1], can be considered according to the following five levels: physical, psychological, social, organizational, and political. Each level presents important considerations for context management systems, as they must make their response adaptations fit both the unfolding real-world and human-factor dynamics.

At the *physical* level an adaptive system should fit the context related to the user's physical body, including its location and functional bio-states (e.g., muscle signals, heart rate, and galvanic skin response). At the *psychological* level the system should fit the internal cognitive view of the user, including the user's mental states, moods, intentions (as far as can be deduced), and particular (or known) cognitive limitations (e.g., working memory load, stress responses, and alertness levels). At the *social* level the system should fit the social-team dynamics of the user's personal network of both devices and independent agent actors (whether other humans or autonomous software entities). At the *organizational* level the system should fit the formal structures in place within the user's organization and respond to the dynamics within that structure, in addition to the role of the organization with other organizations (e.g., when information must be shared between two organizations, the goals of these organizations must be considered). Lastly, at the *political* level the system should fit the current policies in place, enabling specific measures to be taken by the system in response to the dynamics at this level (e.g., accounting for the cascading effect of the introduction or removal of particular communications policies).

Together, these levels and their resulting considerations work to provide a fuller, human-factors view in support of context management in adaptive systems. In the next section, a modular approach to account for the multi-dimensionality of human-factors is proposed for the RECON system and is based on the notion of expert agents, each targeting a unique perspective for system adaptation.

III. A Multi-Agent Context-Management Architecture for RECON

The RECON context-management architecture, while also being concerned with both the explicit and implicit contextual landscape of the user and system, emphasizes flexibility (i.e., being able to use different service providers to achieve the same function) and adaptability (i.e., being able to respond to changes in order to support the user in reducing information overload). To achieve these goals, a four-layer multi-agent systems approach is proposed as a foundation for the context-management architecture. The following four layers, identified in Fig. 1, are described below: the agent communication middleware (A); the generic, expert service-oriented agent community (B); the agent-oriented interaction logic (C); and the application-specific interfaces (D). Combined, these layers create a system in which a community of experts—consisting of both humans and agents—can work together synergistically to reduce information overload and improve organizational effectiveness through emergent adaptation, which is discussed in the final subsection.

A. Agent Communication Middleware

Communication is an essential requirement for a community of flexible agents [10], [11]. Software agents must be able to communicate dynamically with other agents without hard-coding specific agent communication at design-time, thereby necessitating the need for agent discovery at run-time. Such a mechanism is greatly facilitated by existing agent communication middleware, such as can be found in JADE [10], a FIPA-compliant, Java-based multi-agent platform, and SPADE [11], a FIPA-compliant, Python-based multi-agent platform.

A communication channel is provided in this middleware to enable agent-to-agent interaction. In such communication middleware, an agent directory is present in which agents can register to become part of the community. The specific service(s) provided by the agent is also tracked using the service directory. Agents requiring a specific service need only search the directory to find the corresponding agent(s) that provides the service. It is then left to the agent's internal logic to determine which agent service to select.

B. Generic, Expert Service-Oriented Agent Community

RECON envisions a multi-solutions approach toward reducing the problem of information overload, where each independent solution can be combined to form a single, more comprehensive solution. These various solutions, including improved HCI adaptation, BCI monitoring, and data filtering, can be considered systems in their own right, each with a particular "expertise." Each agent can be created generically, without a specific domain or application in mind, having its own particular expertise, and living in a community of other agents that can come together to provide services to a particular application.

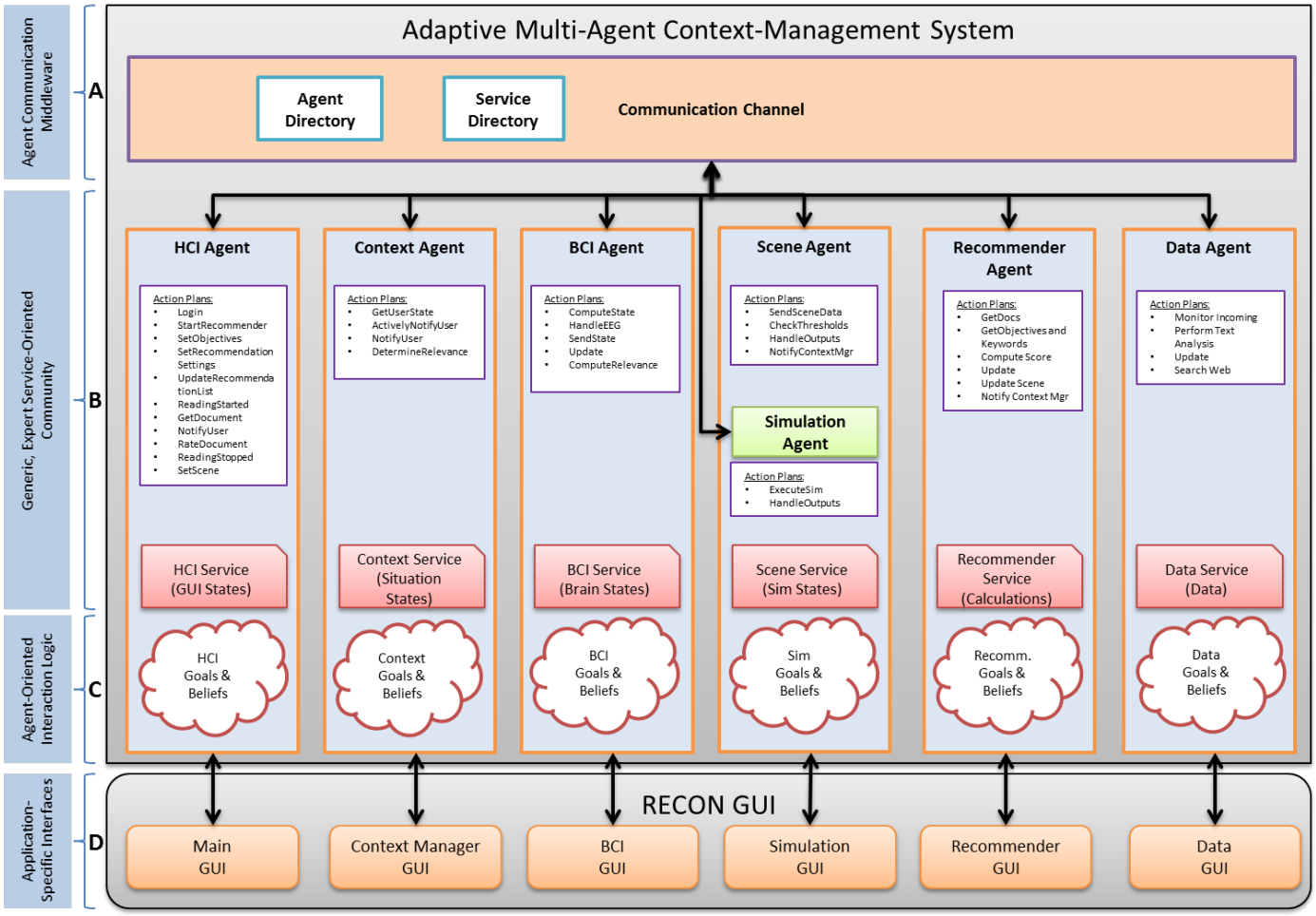

Fig. 1. A four-layer, multi-agent context-management architecture in which a community of generic, expert service-oriented agents provides services to a domain-specific application, RECON, concerned with reducing a user's information overload.

The vision is that this community of agents will increase over time, with the caveat that it is not necessary that all agents participate in a given solution or that there be only one agent that can provide a particular service. However, to begin this community, the following seven service-providing, RECON-relevant agents have been identified as shown in Fig. 1.

1) Human-Computer Interface (HCI) Agent: This agent's role is concerned with monitoring and managing the HCI. Its main actions include identifying the explicit context of the user (e.g., is the user logged into the system, currently setting objectives, or reading a document). It is also responsible for being aware of how a particular application's graphical user interface (GUI) can be adapted (e.g., can specific portions of the display be hidden so as to minimize user distraction).

2) Brain-Computer Interface (BCI) Agent: This agent's role is concerned with monitoring and classifying the psycho-physiological state of the user. The actions associated with this agent's particular expertise include monitoring EEG signals (from the user's headset) and classifying the user's implicit contextual state based on established models of EEG analysis (e.g., measures based on excitement, relaxation, alertness, and stress levels) [12]. This specialization allows this agent to be active only for those applications, like RECON, that make use of brain-state information.

3) Data Agent: This agent's role is concerned with monitoring and analyzing incoming data, and with collecting data from the Web based on specific Web-crawling parameters (e.g., specific website domains and keywords specified by the user). The exact run-time behaviour of this expert agent for a particular application will depend on the specified needs. The actions associated with this agent include collecting and monitoring incoming data (e.g., documents), running this data through different text-analyzing engines (such as AlchemyAPI [13] and OpenCalais [14]), and storing this processed data in a database for later use by other agents.

4) Scene Agent: This agent's role is concerned with the creation and monitoring of *scenes*. A scene refers to a particular aspect of a situation that a user may wish to offload to the system (e.g., perhaps the user is interested in being notified if $> X$ documents are found containing a particular set of keywords). The actions associated with this agent include storing scenes created by the user, monitoring incoming processed data

to determine if specific scene conditions have been met, and issuing a notification (alert) if a scene's condition threshold has been reached.

5) Simulation Agent: This agent's role is concerned with the creation and execution of simulations, whose results act as particular scene conditions. The actions of this agent include storing the location of external simulation models or the models of internal simulations supported directly by the agent, as well as the input parameters that are passed to these simulations (e.g., a real number in the range [0,1] specifying the system's confidence in the relevance of a particular keyword based on the number of recent documents containing this keyword). Other actions include executing the simulations and storing the results in a database, which can be accessed by other interested agents.

6) Recommender Agent: This agent's role is concerned with ranking processed data (i.e., tagged documents) based on specific recommendation criteria so as to present the user with a recommendation of the most relevant system-data available. The actions of this expert agent include storing the user-specific recommendation criteria (e.g., relevant keywords, preferred website domains, and user rating history), and updating the recommendation list based on newly processed documents and user action and feedback (e.g., which documents have been read and what ratings were provided by the user).

7) Context Agent: This agent's role is concerned with assessing the current overall context of the user. The actions of this agent include acquiring all available implicit and explicit context from the other agents within the community, determining the current context of the user, managing what information is sent to the user (e.g., from the Recommender and Scene Agents), and initiating available GUI interventions (via the HCI Agent) to reduce experienced information overload on the part of the user. It is from these other agents that the context agent will collate and make sense of this information based on the active agents (i.e., those which form part of the multi-solutions approach for the specific application under consideration).

C. Agent-Oriented Interaction Logic

As discussed above, an agent will attempt to find other agent services it needs within the community proactively and, thus, self-organize around its goals without any explicit direction within the code. This is accomplished through the use of goal-oriented, belief-based agents. The Belief-Desire-Intention (BDI) software model enables an agent to be deliberative, i.e., the agent can select the action it will perform next based on the current state of the system [15]. This state includes all of the facts within the agent "world" (i.e., system)—incoming documents, HCI logs, BCI data, and simulation results—but that each agent, because of its expertise, restricts its consideration of these facts to those relevant to the completion of its goals. This model also ensures that these agents have bounded rationality so that the specific beliefs of a particular agent are not known to other agents unless they are explicitly shared. Such features enable flexible agent-to-agent interaction. Ultimately, the present architecture allows several agents to offer the same expert service, and an agent can choose which other agent(s) to interact with based on criteria such as past experience and recommendations from trusted "friends" [15], [12]. In this way, agents can be highly flexible, while still maintaining distinct expertise.

D. Application-Specific Interfaces

To link the generic agents within the expert community to a specific domain application an interface is required. This interface acts like a mechanism for articulating the needs of the application while respecting the expert services provided by the agents. Because the agents are both belief-based and goal-based, the articulation of application-specific requirements must necessarily take the form of world facts and agent beliefs. For example, for the Data Agent, beliefs can be provided by the application interface specifying the location of the data repository to monitor, a list of important keywords to search for, and specific beliefs (flags) related to the agent's text analyzers.

There need not be a graphical interface associated with each belief-based agent at the application layer (e.g., in the case of the Data Agent, particular beliefs can be specified in code by the application developer, thus disallowing user-specified modifications). However, for the RECON application, a GUI has been created for each of the agents to increase the ability to tailor the system to a specific user. As the application is being developed to assist the user in reducing the problem of information overload, this may require high levels of customization [16].

E. Towards Emergent Adaptation in RECON

Rather than by the effects of any one agent, it is the combination of the interaction and particular expertise of each agent that enables system-level adaptation to occur [17]. Specifically, in RECON, adaptation is primarily a combination of the following: the HCI Agent, i.e., the interaction log of what the user has done and is currently doing, as well as the explicit situational context and the possible GUI adaptations that have been enabled in code; the BCI Agent, i.e., the current psycho-physiological state of the user and how it compares to the user's "normal" state; the Scene and Recommender Agents, which provide recommendations/notifications to the user; the Context Agent, which must collate this disparate information into action(s) that help reduce analyst overload; and the Data Agent, which influences the number of documents in the system, via its filter, and the meta-information associated with each document, via the text analysis mechanisms it employs. Thus, the adaptation of the system can be said to emerge as a result of the total interactions across agents, based on their individual perspectives of the unfolding situation and their particular action plans (response behaviours).

IV. Cognitive Modelling for Context Management in RECON

As discussed in Section II, the human psychological-level context represents an important challenge in the design of

adaptive systems. However, in addition to detecting the user's psychological states, it is important to be able to operationalize this information in order to improve adaptive system behaviour. In this section, a human-performance cognitive-behavioural model is considered for the purposes of context management in RECON, mapping states to behaviours and system adaptation. Such an approach allows for system-level decisions about possible adaptations that could be beneficial based on the user's particular context.

The COntextual COntrol Model (COCOM), described in [18], is a foundational model outlining four different control states that can be in effect for an individual based on the amount of time remaining for decision-making. These states—strategic control, tactical control, opportunistic control, and scrambled control—represent a continuum from strategic control, where the decision-maker has sufficient time to plan, to scrambled control, where the decision-maker is faced with very limited (to potentially no time) to plan. These control states result in set parameters that can be considered in calculations, such as the estimation of available time to an event horizon (i.e., the time until a significant event occurs) [19].

Similar control-specific notions are also applicable to context management and, in the case of RECON, a natural mapping can be made for the intelligence analysis domain. As shown in Fig. 2, the COCOM model has been fitted to support RECON's context-management approach. In RECON, two cases (or classes) of recommendation exist: (i) scenes, which can include things such as newly simulated situation projections; and (ii) documents, which consist mainly of new input data from the Web. Scenes tend to reflect higher-level, strategic and tactical outlooks and would generally be created only when the user has sufficient time. However, recommendations (or alerts) related to these can be presented by the system at any time (independent of the user's context mode), in much the same way as document recommendations.

A user will be in one of four (possibly overlapping) context modes, determined by the context manager based on both explicit (e.g., HCI logs) and implicit (e.g., BCI state classification) context. These context modes, ordered according to decreasing time-to-plan and directly based on the mapping from COCOM states, are as follows: the *strategic context mode*, where there is a significant amount of time remaining before a decision is required; the *tactical context mode*, where there is sufficient time remaining to consider alternate avenues; the *opportunistic context mode*, where time is limited; and the *scrambled context mode*, where time is very limited (or has run out) and a decision must be made as soon as possible.

These four modes map to system adaptation and recommendation. When the strategic context mode has been identified, the system performs no special filtering of recommendations and alerts, allowing the user to view a wide range of information and case-projections that may not be "on-task." Likewise, when the tactical context mode is deduced by the context manager, the system makes use of low filtering and hence the more off-task alerts and case projections are not directly presented to the user. In the opportunistic context mode, the system uses a medium level of filtering for recommendations and alerts, allowing near-task information to be shown to the user. Lastly, in the scrambled contextual mode, the system adapts with high filtering of incoming recommendations and alerts, presenting only on-task information to the user. The determination of on-task recommendations and alerts is based to a large extent on user preferences, such as the ranking of current objectives, keywords, and scenes, while the determination of the current context mode, as discussed earlier, is based on a combination of user-context data from both explicit and implicit sources. Following a system response, both implicit feedback (e.g., brain state classification) and explicit feedback (e.g., user ratings of documents) are provided to the relevant expert agents within the community to determine the effectiveness of the adaptation, thereby allowing for improved adaptation in the future.

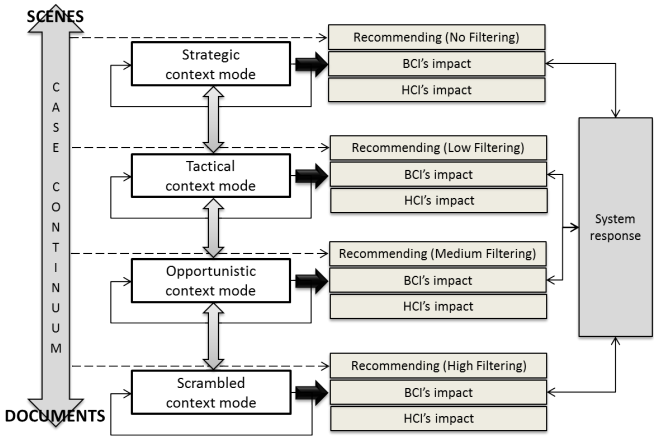

Fig. 2. The COCOM model [18] applied to context management.

V. Related Work on Multi-Agent Systems for Context Management

In this section, a selection of related work on the use of multi-agent systems in context management is presented. These efforts reflect the importance of the context-management problem and the applicability of agent-based approaches.

In [3], the authors describe a multi-agent mobile context-aware system, where humans with client agents on their mobile phones navigate around a city, while the client agent connects to the context agent to determine nearby points-of-interest based on a generic points-of-interest ontology, in addition to finding people with similar interests close by. Such an approach, while multi-agent, does not use multiple agents in determining context, nor does it use a cognitive model for guiding context classification. Instead, it defines context strictly as being a location-based entity.

In [20], the authors propose a multi-agent architecture for achieving user requests in a pervasive computing environment. At its core, it uses three agents to achieve context-awareness: the context collection agent, the ontology agent, and a reasoner agent. In some respects this is similar to the current work, as multiple agents are involved in context management;

however, the specific agents used, as well as the mechanism for ultimately determining context are quite different—for example, they do not use a cognitive model to guide context classification.

In [21], the authors propose a context-management system comprising four agents: a context monitor agent, a configuration selector agent, an enforcer agent, and a visualizer agent. Although a multi-agent approach is employed, this research focuses on the problem of improving software robustness in the face of viruses and malware and does not emphasize a community of generic expert agents.

In [22], the authors propose an ontology-based decision support approach for military information systems. As part of the approach, a multi-agent methodology is described in which five agents work together to support decision-making: a sensor agent, a context-management agent, a decision-support agent, a user agent, and an information-service agent. While similar in terms of domain, this approach does not share the same agents as RECON, nor is the emphasis focused on reducing information overload, although it does target improved decision making. In addition, context appears to be derived only from rules pertaining to the sensor data and not a combination of the data (and, thus, context) arriving from several agents.

These works have shown that, while agent-oriented approaches have had a prominent history in context-aware systems literature [6], the use of a multi-agent approach to improve a system's context-awareness has received relatively less consideration and remains a promising avenue of exploration.

VI. Conclusion

The RECON vision targets the adaptive context-aware systems domain for intelligence analysts, and has an explicit human-factors view of context management. Moreover, it emphasizes the importance of assessing context from a multi-dimensional perspective. Systems that employ HCI, BCI, and simulation in their determination of context are rare, as are systems that rely on a cognitive model as the basis for their context classification. These advances have been combined and form the foundation of the multi-agent context-management architecture extension introduced in this paper. Specifically, in this work, a generic, expert service-oriented agent community was presented alongside a novel application of the COCOM model.

As part of future work, the proposed extension will be integrated within the existing RECON implementation. In addition, a human-in-the-loop, serious gaming experiment is being developed to investigate the effectiveness of this approach in reducing information overload based on the principles of adaptive context management. It is envisioned that such an approach will be suitable not only for the intelligence analysis domain, but also for a broader user group. This will further the goal of futuristic, adaptive software-intensive systems.

Acknowledgment

This work was funded in part by Defence R&D Canada, Thales Research and Technology Canada, and a research partnership grant from the Department of National Defence and the Natural Sciences and Engineering Research Council of Canada.

References

[1] K. Vicente, *The Human Factor: Revolutionizing the Way People Live with Technology*. Routledge, 2004.

[2] W. Ross, A. Morris, M. Ulieru, and A. B. Guyard, "RECON: An adaptive human-machine system for supporting intelligence analysis," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 782–787.

[3] Ö. Yılmaz and R. C. Erdur, "iConAwa – An intelligent context-aware system," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2907–2918, 2012.

[4] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, 1999.

[5] G. Fischer, "Context-aware systems: The 'right' information, at the 'right' time, in the 'right' place, in the 'right' way, to the 'right' person," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, 2012, pp. 287–294.

[6] J.-y. Hong, E.-h. Suh, and S.-J. Kim, "Context-aware systems: A literature review and classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.

[7] D. Gouin, V. Lavigne, and A. Bergeron-Guyard, "Human-computer interaction with an intelligence virtual analyst," *Proceedings of Knowledge Systems for Coalition Operations, IHMC, Pensacola, FL*, 2012.

[8] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with Computers*, vol. 23, no. 1, pp. 4–17, 2011.

[9] A. Morris, "Socio-technical systems in ICT: A comprehensive survey," University of Trento, Tech. Rep., 2009, http://eprints.biblio.unitn.it/archive/00001670/.

[10] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE: A FIPA2000 compliant agent development environment," in *Proceedings of the Fifth International Conference on Autonomous Agents*. ACM, 2001, pp. 216–217.

[11] M. E. Gregori, J. P. Cámara, and G. A. Bada, "A jabber-based multi-agent system platform," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 2006, pp. 1282–1284.

[12] A. Morris and M. Ulieru, "FRIENDs: Brain-monitoring agents for adaptive socio-technical systems," *Multiagent and Grid Systems*, vol. 8, no. 4, pp. 329–347, 2012.

[13] AlchemyAPI, http://www.alchemyapi.com.

[14] OpenCalais, http://www.opencalais.com.

[15] J. Carbo, J. M. Molina, and J. Davila, "A BDI agent architecture for reasoning about reputation," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 817–822.

[16] D. Lafond, R. Proulx, A. Morris, W. Ross, A. Bergeron-Guyard, and M. Ulieru, "HCI dilemmas for context-aware support in intelligence analysis," in *Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE), The Sixth International Conference on*. IARIA, 2014.

[17] W. Ross, A. Morris, and M. Ulieru, "NEXUS: A synergistic human-service ecosystems approach," in *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 175–180.

[18] B. F. Gore *et al.*, "Human performance cognitive-behavioral modeling: A benefit for occupational safety," *International Journal of Occupational Safety and Ergonomics*, vol. 8, no. 3, pp. 339–351, 2002.

[19] A. Morris, W. Ross, M. Ulieru, and P. Chouinard, "Modelling public security operations: Evaluation of the holistic security ecosystem (HSE) proof-of-concept," DTIC Document, Tech. Rep., 2012.

[20] Z. Wei, N. Wang, M. Kang, and W. Zhou, "An agent-based context-aware middleware for pervasive computing," in *Information Science and Engineering (ISISE), International Symposium on*, vol. 2. IEEE, 2008, pp. 116–119.

[21] N. A. Qureshi and A. Perini, "An agent-based middleware for adaptive systems," in *Quality Software (QSIC), The Eighth International Conference on*. IEEE, 2008, pp. 423–428.

[22] S. Song, K. Ryu, and M. Kim, "Ontology-based decision support for military information systems," in *Applications and Technology Conference (LISAT), Long Island Systems*. IEEE, 2010, pp. 1–5.