Architectural Design Considerations for Context-Aware Support in RECON Intelligence Analysis

Alexis Morris*, William Ross*, Mihaela Ulieru†, Daniel Lafond‡, René Proulx‡, and Alexandre Bergeron-Guyard§

*Faculty of Computer Science, University of New Brunswick, Fredericton, Canada
{alexis.morris, william.ross}@unb.ca
†School of Information Technology, Carleton University, Ottawa, Canada
{mihaela}@theimpactinstitute.org
‡Thales Research and Technology Canada, Quebec City, Canada
{daniel.lafond, rene.proulx}@ca.thalesgroup.com
§Defence Research and Development Canada (Valcartier), Quebec City, Canada
{alexandre.bergeron-guyard}@drdc-rddc.gc.ca

***Abstract*—The REcommending Cases based on cONtext (RECON) system is a prototype adaptive technology designed to support intelligence analysts in overcoming the problem of cognitive overload. Its central objective is to assist these analysts during the collection, processing, and analysis phases of the intelligence cycle through sense-making of both explicit and implicit contextual information. RECON combines machine learning, text-analysis, brain-computer interfaces, and simulation to create an innovative case-based recommendation capability. In developing RECON, multiple considerations have been explored based on key human-computer interaction dilemmas that emerge when designing joint-cognitive systems endowed with an adaptive capacity. Herein, eight architectural design considerations are discussed, related to human-modelling, human-machine interaction, and human-machine synergy, which have impacted the system development. The central RECON architecture and its components are also presented, including a context-sensitive cognitive model based on COCOM. This work aims to provide these core architectural components and their design considerations as a contribution toward aiding developers in designing, customizing, and improving future adaptive context-management systems.**

Keywords–adaptive systems; context-awareness; human factors; human-computer interaction; brain-computer interfaces.

I. Introduction

The development of adaptive software systems and infrastructures that are situationally responsive and human-centred remains an attractive area of research, as technologies progress and people continue to work with more data as part of their routine tasks. The improvement of human-centred technologies involves a synergy of both human and machine in order to address the dynamics of unfolding situations; hence, systems that are both dynamic and responsive are required. These dynamic systems are inherently open, interacting with the environment of the organization to carry out its goals, which are often time-sensitive, as in the case of real-time information systems, or even critical, as in the case of emergency response. Moreover, the problem of information overload is becoming increasingly evident in the world, as people become more connected with technology, and this trend is only expected to continue with the proliferation of "big data."

Having technology that can adapt to the varied needs of the user—be they task-related or cognitive-related needs—and having systems that can incorporate humans-in-the-loop to sift through large volumes of data in order to effectively gather and assess information offer direction toward a possible solution to the problem. These point directly to a context-sensitive approach for achieving human-machine synergy, where the combined results of the human and software system working together is greater than the result of any one component working in isolation. The development of such systems requires design considerations that are both human-computer-interface (HCI)-focused and context-based, as discussed in the authors' previous work from ADAPTIVE 2014 [1], which highlights core HCI dilemmas for context-aware support in the intelligence analysis domain.

Adaptation in human-machine systems is challenging, as it requires significant information monitoring. The human must monitor incoming information in order to determine appropriate decisions and response actions, and the technological system must monitor user-context information in order to adapt to the user and perform functions in a dynamic environment. Also, human-machine systems involve the often-complex interplay of human and technological components as interconnected actors sharing a common goal. To be agile, both the human-in-the-loop and the technological system must be in sync with the speed, scope, and context of real-world dynamics, as interactions in a complex real-world situation require corresponding complexity in adaptive systems [2]. However, it is known that these systems, which combine the human-social and technological dimensions, can often become out-of-sync in fast-paced situations where human decision-makers routinely require actions that are outside of the design scope of the technological systems on that they depend [3]. There is a need, therefore, for technology to support users in varied situations that are inherently human-centred, and such a practical adaptive system should enable the user to have balanced access to the most relevant information available (especially in information-centered domains), while also providing this information in a timely manner, in sync with the user's total context.

The current paper extends the authors' previous effort in [1] with a more comprehensive presentation of the architectural design considerations for the developed RECON (REcommending Cases based on cONtext) architecture, which will be presented in detail. Moreover, as these strongly influence the resulting functionality of the developed software system, architectural design considerations should be carefully and explicitly examined and conscientiously applied. Herein, eight key considerations encountered during the course of design and early implementation will be presented and critically discussed. This work promotes these design considerations, which have been accounted for in the development of RECON, as an important step towards the development of future architectures for adaptive, context-aware management, and the expected audience for such design considerations are those involved in the information analysis domain, where cognitive overload is prevalent. The developed RECON system will be used as a case study in how to apply these considerations.

The remainder of this paper is organized as follows. Section II highlights the intelligence-analysis domain, including the problem of cognitive overload and context-awareness. Section III outlines the use case and detailed architecture for RECON. Section IV introduces relevant design considerations for context-aware systems development, along with a taxonomy of such considerations, while Section V presents how the introduced design considerations are applied to the RECON system implementation. Next, Section VI highlights related work on architectural design considerations and compares these with those applied in the RECON case. Lastly, Section VII concludes the paper and offers potential avenues for future work.

II. Problem Domain

In this section, the problem domain is described in more detail, along with the challenge of cognitive overload and a promising path toward a potential solution involving context awareness.

A. *Intelligence Analysis Domain*

To motivate the need for context-aware architectures in the intelligence domain, it is important to highlight the typical information cycle and its effect on the role of the intelligence analyst, as a key player, interested in sense-making and accurate projections for multiple situations that are often time-sensitive and multi-faceted. This intelligence cycle is defined as “the process of developing raw information into finished intelligence for policymakers to use in decision-making and action” [4]. The intelligence cycle encompasses many sense-making tasks that the intelligence analyst must accomplish in an iterative fashion. Such tasks include: gathering relevant information, representing and organizing the information in a schematic way that will ease the analysis process, developing an understanding of the situation by subjecting the information to various hypotheses, and producing intelligence packages and recommendations for courses of action.

As described by Pirolli and Card [5], the overall process of sense-making is organized into two major loops of activities: (1) a *foraging loop* that involves processes aimed at seeking, searching, filtering, reading and extracting information, possibly into some schema [6]; and (2) a *sense-making loop* that involves iterative development of a mental model (a conceptualization) from the schema that best fits the evidence [7]. This process is illustrated in Figure 1.

The intelligence cycle, described in [5], is shown in Figure 2. This cycle includes activities involving planning and direction, collection of data, processing of data, analysis and production of resolutions and projections, and dissemination of information to decision-makers. Here, the intelligence analyst performs the role of seeking out information for a set of unfolding situations, many of which may be dynamic and fast-changing, through collation of multiple documents.

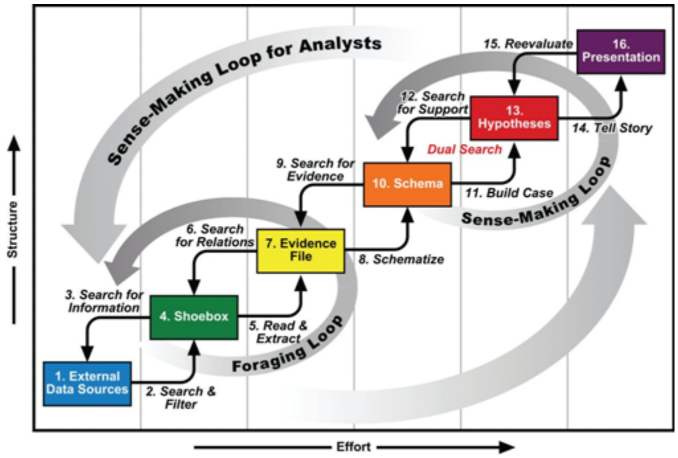

Figure 1. Notional model of sense-making (from [5]).

Figure 2. The intelligence cycle (adapted from [10]).

While the day-to-day activities of the intelligence analyst are driven by this intelligence cycle, the analyst’s activities are subjected to a number of contextual factors (e.g., psycho-physiological and environmental) that can severely impede intelligence analysis due to excessive workload, time pressure, and uncertainty [8]. However, the primary cause of concern is that of cognitive overload, which impacts the analyst’s ability to effectively identify situationally-relevant information due to data overload (i.e., too much data to sift through) and/or cognitive limitations (i.e., too much complexity in the data for making immediate sense without assistive analytical tools) [9]. Together, these present a critical challenge to the development and success of advanced adaptive systems, where humans-in-the-loop must make sense of an ever-increasing inflow of data in order to perform their tasks.

B. *Context Awareness*

To manage the dynamics of real-world information monitoring and sense-making, there are many different 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 [8], *context* is considered as anything that can be used to correctly identify the situation of a user. 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) [11]. Context can also refer to less concretized notions, such as describing users' psycho-physiological states, including their current cognitive mood and stress level. These can be obtained through active and passive sensing of users via bio-metric sensors, but can also be deduced from other sources such as camera monitoring of facial expressions [12].

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" [13]—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). Practical systems that are aware of users with this level of detail are rare, although context-awareness has been a research staple for the past decade [14]. However, such systems are becoming more tenable due to advances in technologies for unobtrusively monitoring users' psychological and physiological states, combined with the technological trends towards miniaturization and improved efficiencies in computational speed and memory costs. As a result, it is now possible to develop better adaptive human-machine systems, synergistically enhancing both human and machine intelligence.

This section outlined the domain of study, namely, intelligence analysis, and the problem under investigation, namely, reducing cognitive overload. It also highlighted the relevance of context-awareness as a promising solution area. These are examined further in the following section, which presents the five-layer, context-aware RECON architecture in detail.

III. RECON: Recommending Cases Based on Context

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 part of Defence Research and Development Canada's iVAC (Intelligent Virtual Analyst Capability) project [15], RECON uses an adaptive-systems approach for information offloading and filtering to assist intelligence analysts. To this end, it focuses on the following three objectives, and in this section both the general use case and resulting architecture for RECON are described:

1) To support the analyst through appropriate visualization and selection of information based on user preferences and real-time brain-state information;
2) To enable the analyst to offload cognitive processing to the system for machine analysis and case-based recommendation; and
3) To alert the analyst through natural interfaces to relevant information based on context.

A. RECON Use Case

In conducting intelligence analysis, analysts must make sense of a variety of information sources (e.g., text documents, webpages, and supplementary GIS data sources). They assess this data according to specific goals and elements-of-interest as dictated by information stakeholders or other supervisors. They must also account for the severity of an unfolding situation, time projections (i.e., time to an event), and constraints in their deliberation strategies. As a result, analysts inevitably incur cognitive pressures, such as fatigue, attention loss, and stress, as they get closer to decision deadlines and perform longer work sessions. The RECON system aims to allow these analysts to offload aspects of this data collation and processing to its automated reasoner, by accepting analyst preferences and making appropriate document recommendations while adapting to real-time brain-monitoring data from the analyst. The general use case for RECON is outlined in Figure 3, along with a system overview diagram.

In this use case, an analyst (outfitted with a wearable and wireless *brain-computer interface (BCI) monitor*) performs the task of sense-making by reading through a number of text-based documents to identify patterns and trends that are applicable to the current situation-of-interest. The analyst logs into RECON using his or her profile information and defines the situation-of-interest in the form of *objectives*, namely a combination of entity and event relationships, through the use of the *HCI interface*. This interface further allows the analyst to input or update system settings and view documents, document recommendations, and scene notifications. A *scene* is an aspect of a situation that the analyst wishes to offload to the system; this can be defined using a combination of keywords, based on active objectives, parameter thresholds, and simulation configurations set by the analyst (e.g., the analyst may be interested in being notified when $> X$ documents are found containing a particular set of keywords). Simultaneously, the BCI headset performs monitoring and classification of the analyst's brain-waves to deduce states that indicate his or her psycho-physiological responses to the task at hand (e.g., whether the analyst is interested in the material within the document, or perhaps is experiencing the negative effects of cognitive overload).

During the course of the analyst's session with RECON, document recommendations are presented to the analyst as part of a document-notification interface, wherein an analyst may inspect each document. The documents listed in this interface are the result of a *recommendation algorithm*, which identifies each document's relevance to the analyst's objectives (defined using keywords and relationships). These documents are retrieved from available repositories, such as online news sites, social media, blogs, and document databases, and are gathered into RECON by a *data text analyzer* that uses existing text-analysis tools to add keyword metadata (used by the recommender) and perform sentiment analysis on the documents. The *context manager*, using its internal logic and the cognitive state of the analyst deduced by the BCI monitor, then determines when to notify the user about new documents and scene-threshold alerts. Lastly, a central *database* stores system-specific information, such as the objectives and preferences of analyst user accounts, enabling the system to continue working toward the defined goals even after the analyst has left for the day.

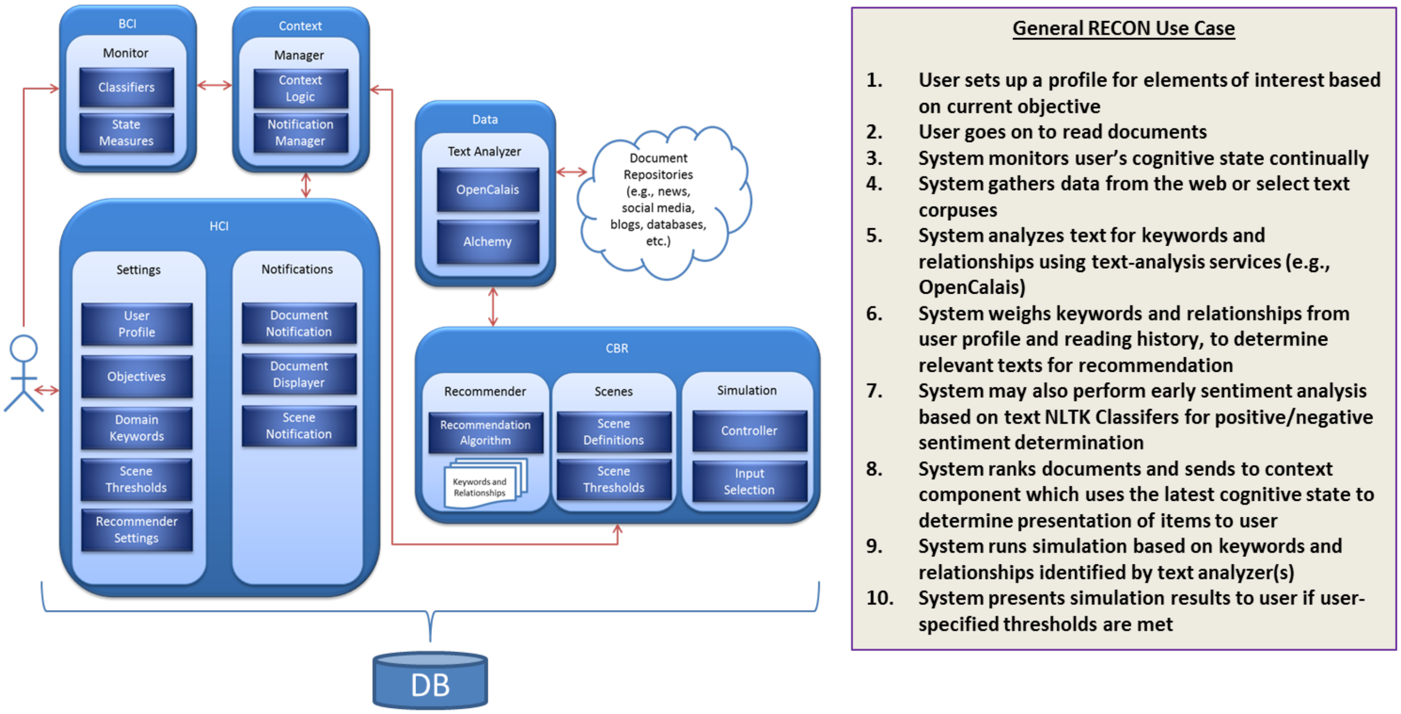

Figure 3. The RECON architecture in more detail: RECON determines context and makes recommendations based on user preferences, brain-state data, and simulation results in order to help alleviate the problem of information overload. The primary use-case is also shown.

B. RECON Architecture

The detailed RECON architecture, outlined previously in [8], is shown in Figure 3. It incorporates the following five layers, which are described below and which correspond directly to the use case.

1) Brain-Computer Interface (BCI) Layer: The BCI layer of RECON is concerned with monitoring and classifying the psycho-physiological state of the user. Its actions include monitoring electroencephalography (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) [16].

In general, BCIs provide mechanisms to acquire, transform, and classify bio-signals into learned states that may then be applied as factors in determining system actions. There are multiple sources of bio-signals related to brain activity that could be useful in real-time, such as EEG, functional Near Infrared Spectroscopy (fNIRS), and functional magnetic-resonance imaging (fMRI); although, for practical purposes, wearability and wireless communication become important criteria for selecting a BCI paradigm. Ideally, the acquisition technology should be as unobtrusive to the user as possible, and not restrict mobility, while simultaneously obtaining brain signals and transmitting these to a processing module. As such, only fNIRS and EEG approaches are relevant solutions, as other techniques involve large and expensive units (see [17], [18] for a discussion on these techniques).

In this work, EEG approaches have been selected as they have the added benefit of being readily available in the form of commercial headsets, including two candidate wireless headsets: the Emotiv EPOC and the Neurosky Mindwave [19], [20]. Whereas the Neurosky Mindwave provides only a single sensor, the Emotiv EPOC has been selected as it provides brain-signal data from multiple sensors, at sites relevant for estimating the states useful for the analyst scenario. In particular, these states include arousal and valence, as in [17], [19], [21]. *Arousal* represents a measure of activation versus inactivation (i.e., being ready to act or not), while *valence* represents a measure of pleasure versus displeasure (i.e., attraction or withdrawal). These have been selected as early measures and can be swapped for new measures, such as alertness and load, as identified in the literature [19], [22]. The combination of arousal and valence provides a circumplex of affect, or emotion, as discussed in [23], which in RECON allows the system to deduce whether analysts are in states such as alert, happy, content, bored, or angry and thereby determine an appropriate response action.

The process whereby signals are translated from raw EEG into state classifications involves the following phases: i) acquisition of raw EEG; ii) pre-processing and noise reduction using discrete wavelet transforms; iii) EEG feature extraction, according to known formulae based on work such as [19], [22]; and iv) classification of features into states, using a classifier trained on labelled datasets (such as [24]) to output levels of arousal and valence.

2) Human-Computer Interface (HCI) Layer: The HCI layer is concerned with monitoring and managing the RECON interface. Its main actions include identifying the current task of the analyst (e.g., whether the analyst is logged into the system, currently setting objectives, or reading a document) and adapting the graphical user interface (GUI) (e.g., whether specific portions of the display should be hidden so as to min-

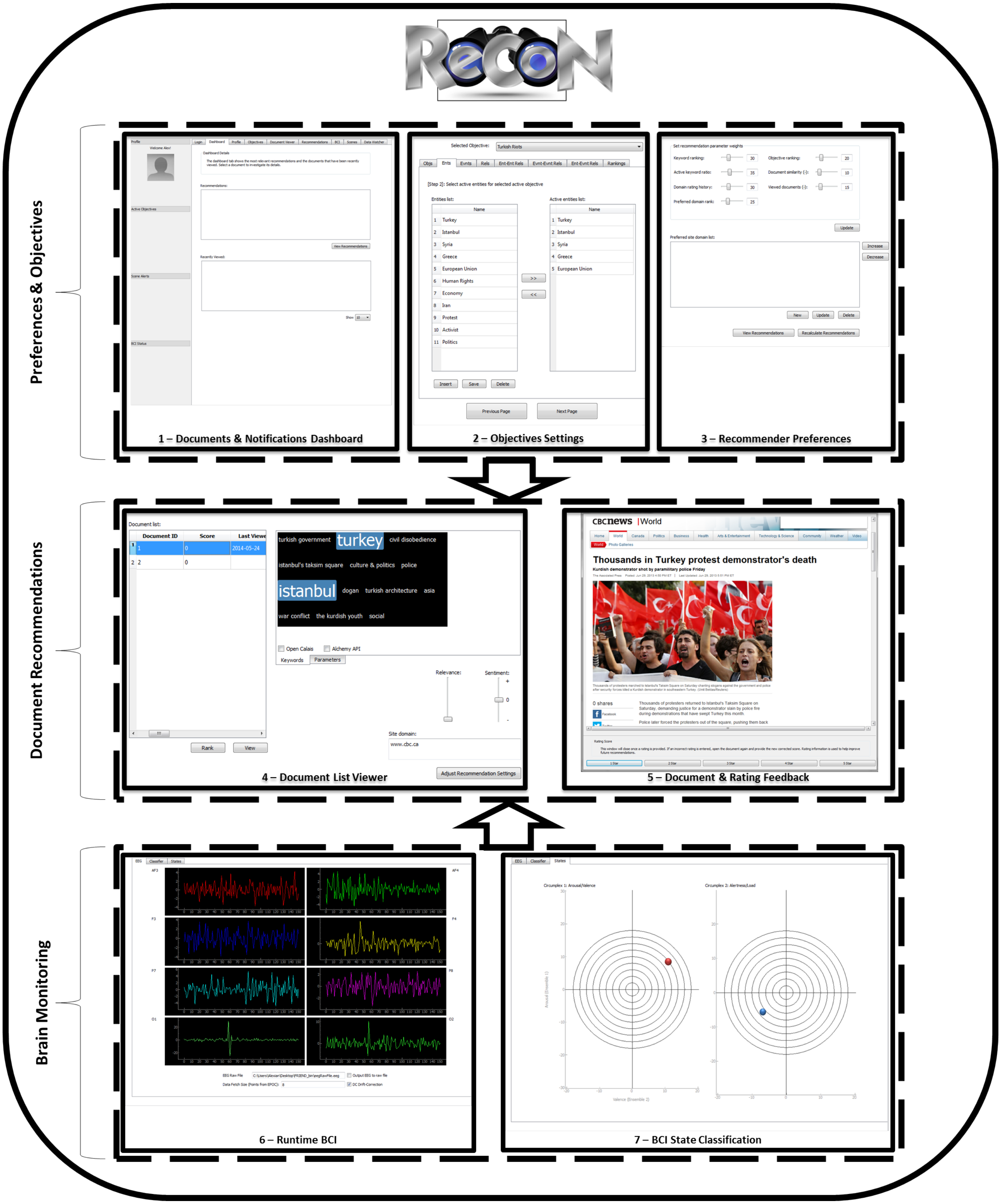

Figure 4. The main RECON interfaces.

imize analyst distraction). This layer also provides the analyst with a means to read and rank documents, set objectives and preferences, and interact with other RECON components, as outlined in the use case from Figure 3.

The main RECON interfaces are shown in Figure 4. These include GUI elements for the following: i) ranked document lists and notifications dashboard, which allows the analyst to view the highest-ranking documents (for both individual and team recommendations) and any system notifications; ii) objectives settings, which allow the analyst to specify and rank his/her target objective(s) in terms of entity and event keywords and relationships; iii) recommender preferences, which allow the analyst to fine-tune the recommendation parameters, such as the analyst's preferred website domains; iv) document viewing and v) ratings feedback, which allow the analyst to view all document recommendations for active objectives, browse document metadata, open documents for viewing, and provide a relevance rating for each document the analyst opens for reading; and vi) BCI runtime and vii) states, which allow the analyst to view the EEG inputs and the corresponding output states. In addition, support also exists for the following GUI elements (omitted from the figure for space reasons): user and team profile, which allows the analyst to set profile information and manage team members; and BCI setup, which allows the analyst (or an administrator) to specify the active BCI classifiers and start or stop real-time EEG monitoring. Together, these interfaces represent the visual and interactive components of RECON.

3) Data Layer: The data layer is responsible for collecting and monitoring incoming data (e.g., documents), analyzing them, and storing the results in a database for later use. This layer monitors preset bodies of textual data, in the form of documents, from websites and other corpuses. When new documents arrive, they are processed using existing text analysis engines, such as AlchemyAPI [25] and OpenCalais [26], in order to provide tagging of documents (e.g., keyword metadata) and sentiment markers. This approach is modular, and specific text-analysis engines, with different underlying ontologies, may be substituted based on the specific domain needs of the active situation(s)-of-interest.

4) Case-Based Recommender (CBR) Layer: The CBR layer is concerned with ranking processed data (i.e., tagged documents) based on specific recommendation criteria so as to present the analyst with a recommendation of the most relevant system data available. The actions include storing the analyst-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 analyst). It also provides an algorithm for document recommendation based on keywords and relationships, the facility for defining scenes and monitoring scene thresholds, and a simulation controller and input-selection mechanism for managing simulations. These simulations are used as scene conditions to model aspects of the situation-of-interest (e.g., system dynamics can be used to estimate particular threshold variables over time, governed by causal-loop diagrams that incorporate variables and their interrelationships, linked to specific objective keywords [27]).

The algorithm developed for the recommender, depicted in Figure 5, makes use of three separate subroutines to calculate the latest recommendation score for all documents in the system pertaining to a specified date range and user. The *getRecommendations* function sets the analyst-specified date range and queries the database checking for i) updated or new analyst-specified objectives (case a) and ii) new documents (case b). If any changes have been made to the objectives (i.e., case a), such as the addition of a new keyword, the *gatherDocumentProperties* function is applied to all documents in the date range; however, if no objectives have been added or changed but new documents are found (i.e., case b), the function is applied only to the new documents. If neither case occurs, the *calculateRecommendationScore* function is called directly. This prevents the system from having to recalculate fixed document properties each time the recommendation algorithm is used. The *gatherDocumentProperties* function retrieves the relevant documents (based on the applicable case), and, for each document, gathers the fixed document properties and compares the document keywords to the keywords and relationships associated with the current active objective(s) of the analyst. The *calculateRecommendationScore* function is then applied. This function retrieves the document properties stored by the previous function and applies analyst-specified modifiers as weights that impact the resulting recommendation score for each document. Such a multi-step mechanism allows the latest recommendations to be computed, taking into account the analyst's most recent ratings and preference settings, without needing to recompute fixed document properties each time a recommendation update is requested.

Specifically, for each document-objective pairing, a document score, σ_{doc}, is calculated according to the following equation:

$$\sigma_{doc} = \phi_{key} * \pi_{key} + \phi_{rel} * \pi_{rel} + \phi_{site} * \pi_{site} \qquad (1)$$

where ϕ_{key} represents the ratio of keywords in the document compared to the total number of keywords specified in the objective; ϕ_{rel} represents the sum of the relevance of matching keywords as ranked in the objective; if the document comes from a site domain that is preferred by the user, ϕ_{site} represents a positive value based on the relative rank position of the preferred site domain within a user-specified list (zero otherwise); and the π_* values represent the preference weighting of each factor (a real number between zero and one inclusively) as specified by the user.

This score is then used in the determination of the document's resulting recommendation score, σ_{rec}, calculated according to the following equation:

$$\sigma_{rec} = \sigma_{doc} + \phi_{obj} * \pi_{obj} - \phi_{sim} * \pi_{sim} - \phi_{acc} * \pi_{acc} \qquad (2)$$

where σ_{doc} represents the fixed document properties according to the equation above; ϕ_{obj} represents a positive value based on the relative ranking of the active objective compared to all active objectives; ϕ_{sim} represents how similar the current document is compared to all documents viewed already by the user in reference to this objective; ϕ_{acc} represents whether or not the document has already been accessed (i.e., viewed) by the user for this objective: one if true and zero if false; finally, the π_* values represent once again the preference weighting of each factor as specified by the user. By reducing the score based on document similarity, the aim is to increase document *coverage* [28] using the following heuristic: recommend to the user the highest scoring documents that have the least similarity compared to documents already viewed.

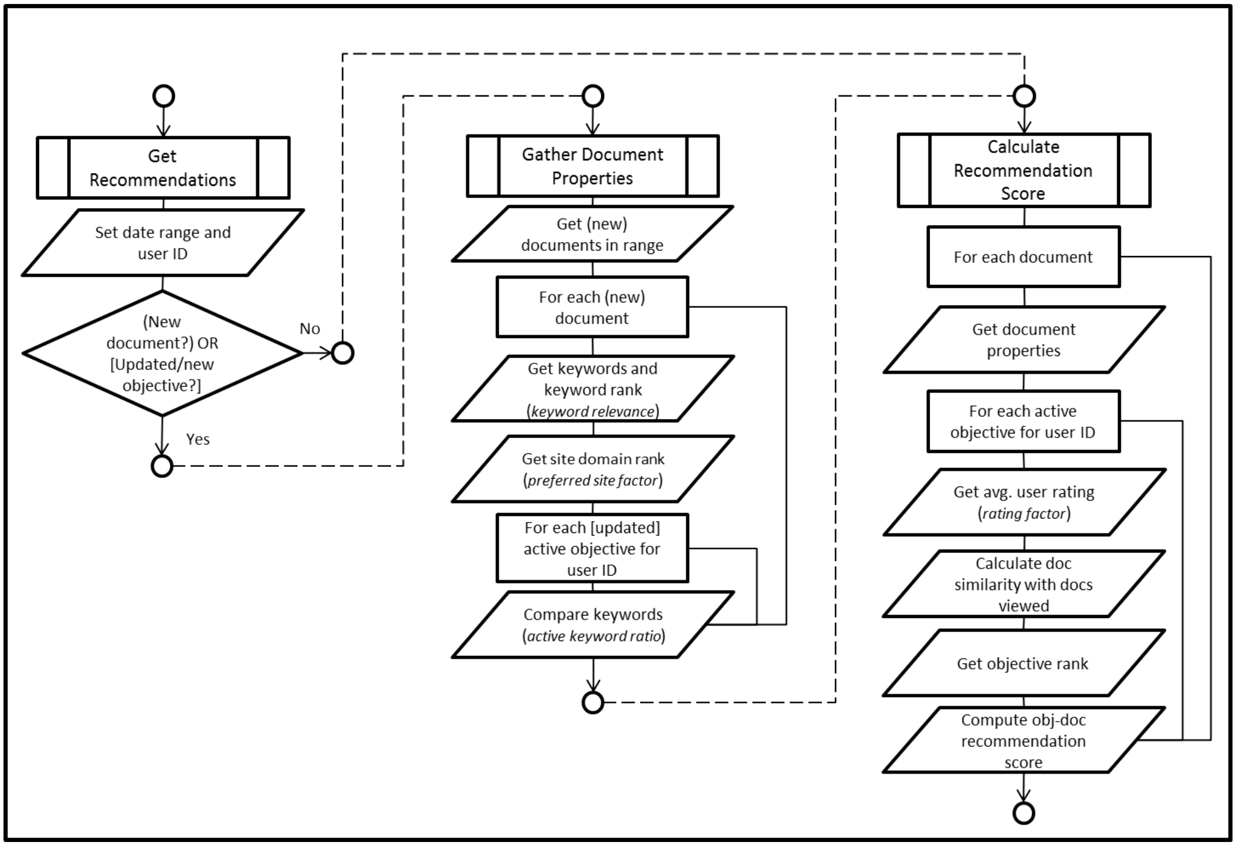

Figure 5. The algorithm used to make document recommendations in RECON.

Moreover, the CBR component is supported by the scene and simulation components, which, together, manage scene notifications coming from the system to the analyst. The scene component is concerned with the creation and monitoring of scenes. The actions of this component include storing scenes created by the analyst, monitoring incoming processed data to determine if specific scene conditions have been met, and issuing a notification if a scene's condition threshold has been reached. The simulation component, on the other hand, is concerned with the creation and execution of simulations, whose results act as particular scene conditions. The actions of this component include storing the location of external simulation models or the models of internal simulations supported directly by the component, as well as the input parameters that are passed to these simulations. This component supports a combination of simulation paradigms (e.g., system dynamics, discrete-event, and multi-agent simulation) to better match the representational requirements of the current situation (e.g., system-level or individual-level concerns) with the most appropriate paradigm [27]. Other actions include executing the simulations and storing the results in the system database.

5) Context Layer: The context layer is the final layer and is concerned with assessing the overall current context of the analyst. The actions of this layer include acquiring all available implicit and explicit context from the other layers, determining the current context of the analyst, managing what information is sent to the analyst (e.g., from the recommender layer), and initiating available GUI interventions (via the HCI layer) to reduce experienced information overload on the part of the analyst. It is from these other layers that the context layer collates and makes sense of this information.

In addition to detecting the user's contextual states, it is important to be able to operationalize this information to improve adaptive system behaviour. Consequently, the context layer makes use of a well-known cognitive model for the intelligence analysis domain as part of its adaptation strategy. The COntextual COntrol Model (COCOM), described in [29] and based off the work of Hollnagel [30], is a foundational model outlining four different control states that can be in effect for an analyst based on the amount of time remaining to make a decision. 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, when applied to the sense-making loop in Figure 1, result in a set of parameters that can be used in determining the analyst's cognitive mode in light of an unfolding event.

As shown in Figure 6, the COCOM model has been fitted to support RECON's context-management approach. In RECON, two classes of recommendation exist: (i) documents, which consist mainly of new input from text sources; and (ii) scenes, which can include things such as newly simulated situation projections. Together, these represent the two "cases" that are recommended by the system according to the current state of the analyst. While documents tend to provide information on a particular situation-of-interest that is more specific in nature

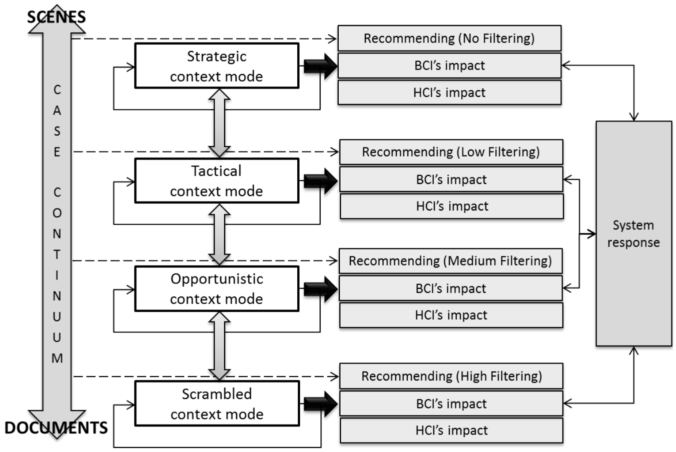

Figure 6. The COCOM model applied to RECON context management (adapted from [29], [30]).

and may arrive at any time, scenes tend to reflect higher-level, strategic and tactical outlooks that would generally be created only when the user has sufficient time. However, alerts related to these scenes can come at any time (independent of the user's context mode), in the same way as document recommendations do.

An analyst can be in one of four context modes, determined by the context layer using both explicit and implicit contextual sources. These context modes, ordered according to decreasing time-available-to-plan and directly based on a mapping from COCOM control 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. Moreover, these four modes map to system adaptation of recommendations and alerts. When the strategic context mode has been identified, the system performs no special filtering of document recommendations and scene alerts, allowing the analyst to view a wide range of information and scene-projections, some of which may not be "on-task." Likewise, when the tactical context mode is deduced by the context layer, the system makes use of low filtering, whereby more off-task alerts and scene projections are not directly presented to the analyst. In the opportunistic context mode, the system uses a medium level of filtering for recommendations and alerts, allowing only near-task and on-task information to be shown to the analyst. Lastly, in the scrambled contextual mode, the system adapts with high filtering of incoming recommendations and alerts, presenting only on-task information to the analyst. The determination of on-task recommendations and alerts is based to a large extent on an analyst's 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 analyst-context data from both explicit and implicit sources.

This section has presented the RECON use case and system in detail, and it is envisioned that such a unique combination of layers, 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 in Section II, thereby improving the machine's ability to effectively assist the analyst and reduce cognitive overload. RECON furthers the goals of alleviating human-cognitive overload in two ways. First, it does so by developing a system capable of sensing and classifying the user's contextual state, including brain state using a brain-computer interface. Secondly, it does so by adapting to the user's context and recommending relevant information to the user based on the system's level of context-awareness. While the vision and use case for RECON have been presented and a proof-of-concept system implemented, they remain to be validated experimentally. However, the foundation for each component is empirically supported by recent literature. In particular, for the brain-computer interface, work such as [19], [31], [32] demonstrates that real-time brain-state classification is indeed viable. In terms of human-computer interfaces, work such as [33], [34] highlights the benefits of integrating HCI and BCI for adaptive systems. Lastly, in terms of both context and recommendation, studies such as [35], [36] underscore the effectiveness of context-based recommendation. These research foundations enable a merger of technologies as presented in RECON, and such a merger requires new architectural design considerations in order to achieve a more cohesive software system. These considerations are examined in the following section.

IV. Architectural Design Considerations

This section outlines eight key architectural design considerations, relevant to the design of adaptive systems at large. These have been grouped into three categories—human modelling, human-machine interaction, and human-machine synergy—according to the taxonomy shown in Figure 7. This taxonomy is described below, followed by a critical discussion of the eight considerations.

A. Considerations Taxonomy

The *human-modelling* category of the taxonomy, shown in Figure 7, relates to design considerations affecting how the human is modelled within the computer system. As a key component of human-machine systems, human modelling acts as the mechanism used by the machine to better understand and represent the user. Two relevant considerations are considered in the next subsection: *model selection*, relating to how user mental states are determined; and *model calibration*, relating to how these specific models are initialized and tuned.

The *human-machine interaction* category refers to the design considerations involving human-machine interaction. These relate to the interface between the human and the machine, with a particular emphasis on the human-in-the-loop acting as a critical component of the overall system [37]. Three considerations are examined in the subsequent subsection: *model transparency*, relating to the extent to which the user understands (and needs to understand) the internal mechanisms driving the system; *user feedback*, relating to how the machine system receives feedback from the user; and *contextual inputs*, relating to how the system receives or collects contextual input from the user (i.e., implicitly or explicitly).

Lastly, the *human-machine synergy* category deals with those considerations affecting the effectiveness of the human-machine team in accomplishing the overall goal of the system.

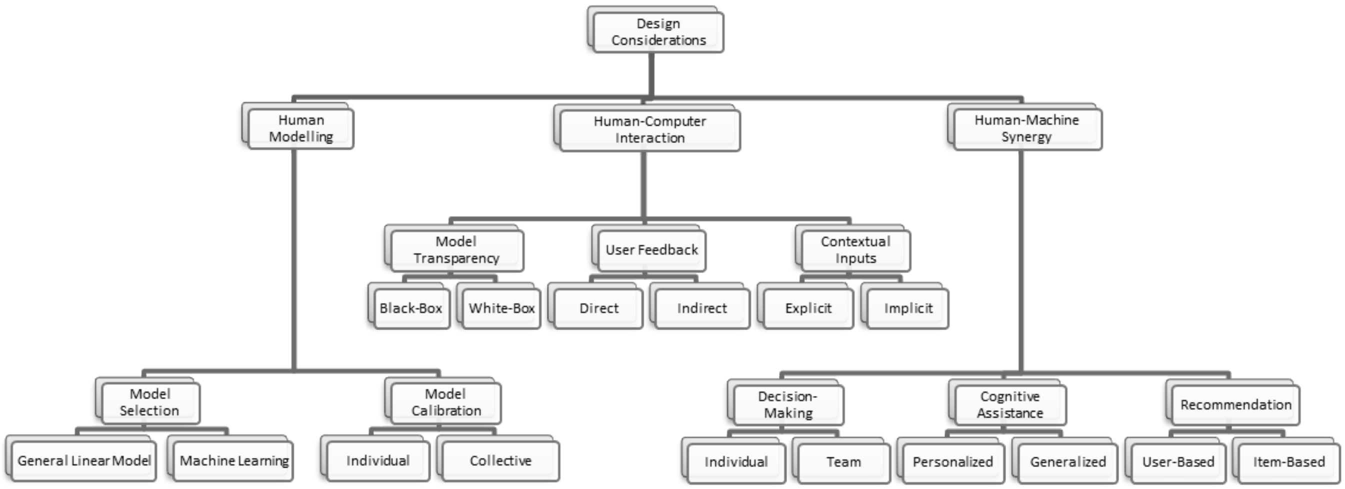

Figure 7. Taxonomy of architectural design considerations for context-aware systems.

Three particular considerations are examined in the following subsection: *decision-making*, relating to the support for organization-wide decision-making within the system; *cognitive assistance*, relating to the the extent of personalization supported within the system; and *recommendation*, relating to the mechanism(s) used in suggesting different information items to the user.

B. The Eight Considerations

The eight architectural design considerations are discussed in detail below and are presented in the order in which they appear in the taxonomy, left to right.

1) Model Selection: General Linear Model vs. Machine Learning: The first consideration, relating to modelling the user's brain state, is concerned with the selection of an appropriate method for capturing the underlying pattern of cerebral activity associated with a given state, namely statistical analyses based on the General Linear Model (GLM) or Machine Learning (ML) algorithms. The GLM approach has a proven track record among the neuroscience community and has robust analysis software available [38]. However, complex non-linear relations cannot be "discovered" using this method (i.e., the underfitting problem) due to the linearity constraint, yet this contraint makes the GLM very robust to noise (e.g., measurement error and intrusions from confounding factors), thus minimising the overfitting problem [39]. Underfitting occurs when the model lacks sufficient functional flexibility to capture a phenonemon, while overfitting occurs when the model's flexibility allows it to "fit" both the true regularities in the data as well as false, noisy patterns, leading to an overestimation of the model's accuracy [40]. ML algorithms, on the other hand, including those related to data mining, provide highly flexible models capable of discovering complex patterns in datasets. However, this flexibility raises a potential vulnerability to overfitting, which must be considered.

2) Individual Calibration vs. Collective Calibration: The second consideration involves model calibration, which can occur at either the collective level or at the individual level. The former results in a single model for all potential users, while the latter results in a distinct, customized model for each user. Individual modelling has the disadvantage of requiring additional overhead for calibration, including a separate data collection for each user in order to extract an individualized model. Nevertheless, this approach may be essential for attaining high levels of model accuracy, particularly in cases when the average of the collective is the result of idiosyncratic patterns [41], [42]. Alternatively, individual differences can be treated as noise, although this could potentially lead to underfitting of the user state.

3) Model Transparency: White-Box vs. Black-Box: The third consideration, related primarily to human-machine interaction, is how much of a model's inputs, logic, and resulting assessment to display to the user. A transparent, "white-box" model may increase user trust in the system, but there is also the risk of fostering mistrust in situations where the user disagrees with the model or does not understand it. Furthermore, a significantly complex display could adversely impact the understandability of the model to the user. On the other hand, a "black-box" approach, in which model details are completely hidden from the user, may also foster doubt and mistrust in the system. As such, this consideration relates to the classic invisibility dilemma: balancing between minimizing distractions from the primary task of the user and providing added value through explicit interaction with the model [43].

4) Direct Feedback vs. Indirect Feedback: The fourth consideration involves whether or not to collect feedback directly from the user in order to improve the underlying models used in adaptation. Direct feedback incorporates the user in the learning process by requiring a manual response about the performance of the system, which in turn guides the fine-tuning of model parameters. This feedback assumes a level of expertise on the part of the user. Moreover, the frequency of direct feedback must be considered, as it may unnecessarily burden the user if required too often [44]. Conversely, indirect feedback allows the system to acquire the necessary inputs for the fine-tuning parameters without involving the user directly. This has the benefit of allowing the user to remain on task, while simultaneously allowing the system to improve its adaptation, as often as needed. There is a tradeoff, however, in terms of the accuracy of the learning mechanism, as some aspects may not lend themselves readily to being deduced indirectly.

5) Explicit vs. Implicit Contextual Inputs: The fifth consideration involves knowledge about user context, which is central to system adaptation. This context evolves according to events and changes occurring during system operation either by explicit interactions from the user (e.g., a user manually indicates current context parameters such as time pressure) or implicit interactions based on the situational context (e.g., automatic data monitoring and sensor-based classification). Explicit context affords the user a sense of control over the system and provides contextual data that may not be otherwise available. However, a system that relies too heavily on this type of context may add to the workload of the user, in terms of providing a larger amount of information manually to the system, and may require a more complex graphical-user interface and additional tasks that may interfere with the user's ability to focus on the task-at-hand. Conversely, a system that emphasizes implicit context frees the user from tedious data input operations, but requires the system to automatically monitor data and perform reasoning to infer the user's contextual information. This demands a significant a priori development effort for effective user-state and contextual classification models.

6) Individual vs. Team Decision-Making: The sixth consideration relates to the manner in which decision-making is performed, and how this can be assisted through technology. In some organizations, individual contribution is valued more than the collective, if not explicitly then implicitly through their reward structure; however, in dealing with complex systems, a broad range of expertise should be drawn upon [45], [46]. Moreover, if people are tired or overloaded, being able to offload a particular task to a more alert member of the team can help the organization make more effective use of its resources [46]. In fact, having the ability to promote a networked culture is seen as a vital step in addressing complex issues [45]. This is not so much a technical issue, as it is an organization-design issue. However, technology can be brought to bear to facilitate or promote the spread of this culture, and such considerations form an integral aspect of system design [47].

7) Personalized vs. Generalized Cognitive Assistance: The seventh consideration, dealing with the human factor, relates to the individual needs of a user. Not everyone is the same, and people have different cognitive abilities and assistance needs. Sometimes people may remember a lot of information at once and be able to recall it; other times they may wish to offload some of this information to a machine. Moreover, each person may view the situation from a different perspective, or have different sub-problems to address. As such, a certain level of user customization may be desirable. However, tradeoffs must be considered. For example, the cost of such customization may be seen as being too high at times. This can be from the point-of-view of the system designer, as it provides more freedom to the user and less predictability on the part of the system [47], but also on the part of the user, who may not see or understand the value in customization. If customization is desired, a possible solution to the latter problem is to provide tutorials and walkthroughs to help guide users in better understanding the benefits of customization.

8) User-Based vs. Item-Based Recommendation: The eighth consideration involves the mechanism used to rank recommendation items. Traditionally, there are two main approaches to recommendation. The first, known as collaborative filtering, involves creating a user profile and comparing it to the profiles of other users. The objective is to find a subset of closest neighbours, whose preferred items can then be recommended to the user under consideration. This is good for situations in which having "trusted friends" can be of benefit to the recommendation (e.g., when wanting to be given a recommendation for a book or movie); however, it does suffer from the cold start problem in which establishing an accurate profile for a new user takes time and many rating samples. The second approach, known as item-based filtering, uses the properties of the items themselves. The objective here is to find similar items to those items the present user ranked most highly. The benefits of this approach are that newer items have just as much chance of being selected as older items and it is good when the set of other system users is small. However, items must be comparable, so it might not work as effectively for recommendations involving a wide-range of differing items (e.g., Amazon). Hybrid methods have also been proposed [28] in which combinations of different recommendation algorithms are used in tandem. These have the effect of mitigating the weaknesses of any one approach, and different techniques may be more suited to specific domains.

This section introduced eight important architectural design considerations for supporting context-awareness in human-machine systems. This together with the previous section, which detailed the RECON proof-of-concept software system, are the focus of the following section. Specifically, it examines how the proposed key architectural considerations have been applied in RECON.

V. Applying the Architectural Design Considerations to RECON

In this section, the application of the eight architectural design considerations presented in Section IV is described according to the five layers of RECON presented in Section III: namely, brain-computer interface (BCI), human-computer interface (HCI), data, case-based recommender (CBR), and context layers. The strengths and benefits of the RECON architecture, resulting from the conscientious application of these considerations, are also discussed, along with possible improvements.

A. Brain-Computer Interface (BCI) Layer

The following design considerations apply to the BCI layer and are presented according to the taxonomy in Figure 7:

- *Model Selection:* A machine learning approach has been selected to recognize dynamic, non-stationary EEG signals. In particular, the use of a neural-network (neuro-fuzzy)-based classifier approach provides a method for making sense of brain EEG data, which is well-supported in literature and provides a generalizable approach to classification, allowing additional state measures to be incorporated [48], [49].
- *Model Calibration:* All selected models are initially calibrated using a collective approach with labelled, pre-existing datasets for training the classifier. This means that there is no need for a lengthy training

session by the analyst prior to using the system. However, there is potential to adapt the classifier to individual characteristics based on performance feedback provided to the system over time.

- *Model Transparency:* A black-box approach to classification has been selected for pattern-recognition, which means that specific details are hidden from the analyst. However, EEG inputs, feature configurations, and state measures can be inspected. This hides low-level model details, which should only be modified by an expert, while providing an overview of the BCI process, which may facilitate the user's trust in the model's output.
- *User Feedback:* An indirect feedback strategy has been selected for the BCI layer, as direct user feedback is obtained elsewhere in RECON and can be applied to the BCI in terms of performance-based adaptation. The benefit of this approach is that the analyst need not have BCI-specific expertise to update the BCI model; instead, this can be abstracted to other parts of the system.
- *Contextual Inputs:* Implicit contextual inputs, in the form of EEG signals, have been selected to deduce an analyst's cognitive state. This allows the system to unobtrusively monitor the user, without requiring explicit input by the user with regards to their current mental state. This is important as the user might not be aware of their own current psycho-physiological state.

B. Human-Computer Interface (HCI) Layer

The following considerations apply to the HCI layer:

- *Contextual Inputs:* Both explicit and implicit contextual inputs have been used in this layer. In particular, explicit inputs have been selected to allow users to set current objectives and preferences, which are used for recommendation and filtering. This has the benefit of allowing the system to know exactly what the analyst is trying to accomplish. Implicit context is also gathered by the system to determine the current task of the user (e.g., whether the user is logged into the system, currently setting objectives, or reading a document), which can be used to adapt system notification levels.
- *Decision-Making:* Functionality for both individual and team decision-making has been supported within the HCI layer. This provides the facility for analyst teams to share objectives and recommendation results and manage team membership, thereby supporting shared situational awareness.
- *Cognitive Assistance:* Functionality for personalized cognitive assistance has been provided in the form of adaptable document lists and notification areas within the HCI layer. This has the benefit of streamlining the presentation of content based on the analyst's current cognitive state and specified preferences (e.g., how the ranking should occur).

C. Data Layer

The following consideration applies to the data layer:

- *Model Transparency:* A black-box approach has been selected for this layer. This is because existing solutions, which are readily (and even freely) available, themselves are black-box solutions, and the development of new approaches to text analysis is outside the scope of the current project. While the specific mechanisms driving a text-analysis engine may not be relevant to an analyst, the ontologies are. As such, within RECON, the results from different text-analysis engines can be activated by the user to compare which is performing best for a particular objective.

D. Case-Based Recommender (CBR) Layer

The following considerations apply to the CBR layer:

- *Model Transparency:* A black-box approach has been selected for recommendation. This has the benefit of hiding the low-level implementation details of the algorithm from the analyst. However, the analyst is still enabled to provide inputs in the form of objectives and tuning preferences that influence the recommender. While the algorithm has been implemented, further testing is required, which may necessitate modifications in order to achieve recommendations that are both relevant and highly diverse when compared to documents the analyst has already seen (as per [28]).
- *User Feedback:* Direct user feedback has been selected for the recommendation layer. The analyst is required to rate each recommended document he/she reads in terms of its relevance to the associated active objective. This has the benefit of allowing the system to obtain immediate and targeted feedback, without unduely burdening the user. This feedback is used to improve future recommendations and can also be used to provide indirect performance feedback to the BCI layer (e.g., correlating BCI state measures with the most-recently rated document to determine a relevance measure for the analyst).
- *Contextual Inputs:* Explicit inputs have been selected for specifying the current situational context of the analyst. This involves setting and defining the current active objective(s), which include the specification of entity and event keywords and relationships between these keywords, along with the relative ranking, used for prioritization, of the objectives and objective components. RECON guides the analyst in defining his/her objectives, and recommender effectiveness is determined precisely by how well the recommendations align with these objectives. The benefit of explicit, guided contextual input is that the system obtains the problem situation directly from the analyst, without having to deduce such potentially-complex and diverse context implicitly. Moreover, this allows for targeted recommendations that can later be refined by modifying the user-specified context.
- *Decision-Making:* Both individual and team decision-making have been selected for the recommendation layer. This means that an analyst can view recommendations related to both his/her objectives, as well as those shared by the analyst's team members. Together, these have the benefit of supporting increased

situational awareness across the team or organization, while still effectively catering to the particular interests or needs of individual analysts. The currently implemented approach could be improved by allowing further refinements in terms of what exactly is shared to other team members (e.g., only share recommendations above a certain rating threshold); however, the team recommendations list can be sorted according to the recommendation rating, among other properties, allowing the analyst to quickly filter items in the list.

- *Cognitive Assistance:* Personalized cognitive assistance has been selected for this layer. This is in the form of scenes. A scene can be defined by the analyst in order to represent a particular aspect of the problem situation he or she wishes to offload to the system. The benefit of this approach is that the analyst is free to define (or not) as many scenes as may prove beneficial. This degree of customization allows expert users to capitalize on personalized cognitive assistance, which can include keyword tracking and multi-paradigm simulations for what-if analysis.
- *Recommendation:* Item-based recommendation has been selected for this layer, where the "items" in this case refer to documents. This approach uses the features of the document (such as tagged keywords), rather than properties of other users who may have read the document, in order to determine the document's relevance to the current analyst. In the traditional approach to item-based recommendation, the properties of other items the analyst has viewed and ranked would be used in the relevance calculation [28]. However, in RECON, because an analyst specifies precisely what he or she is interested in at the present time through objectives, this explicit context is used instead. The benefit is that the analyst's most-recent intentions are always incorporated into the RECON recommendations, rather than using potentially outdated intention history, as is the case in the traditional approach.

E. Context Layer

The following considerations apply to the context layer:

- *Model Transparency:* A white-box approach has been selected for the context layer. The COCOM model has been adapted for context management, and the application of this well-known method allows analysts to better understand the system behaviour (e.g., in terms of its filtering actions). This has the benefit of promoting confidence in the system's behaviour, while also allowing for potential future improvements involving direct feedback from the user with regards to the context mode determination of the system (e.g., the analyst feels he or she is in the tactical context mode, while the system has determined that the current mode is scrambled context).
- *Contextual Inputs:* Both implicit and explicit contextual inputs have been selected for this layer. Implicit context is obtained from the BCI component as well as from the HCI activity log, which shows the current action the analyst is performing in the system. Explicit context is obtained primarily through the analyst's definition of objectives. The benefit of a combined approach is that context that can be deduced by the system (e.g., current psycho-physiological state) is acquired without direct involvement of the analyst, while context that is more difficult to ascertain automatically (e.g., changing objectives and priorities) can be acquired explicitly from the analyst. This promotes an effective balance between explicit analyst involvement in the context adaptation process and system usefulness, allowing the analyst more time to focus on important tasks such as sense-making.
- *Decision-Making:* Currently, individual decision-making has been selected for this layer. This comes in the form of contextual mode classification at the analyst level, which determines how recommendations and alerts are filtered to the individual. A future improvement would be to support team decision-making at this layer. This would take the form of recommendations and alerts being filtered across a team of analysts, where a particular notification would be sent to the analyst best-suited to receive it (e.g., an analyst who is not determined to be overloaded and for whom the content of the document is meaningful, i.e., it matches with at least one of the analyst's individual or team objectives). This would have the benefit of sending the right information to the right person at the right time, three key criteria of the five "rights" discussed in Section II.
- *Cognitive Assistance:* A generalized cognitive assistance approach has been selected for the context layer. This comes in the form of the COCOM-based model, which defines four possible contextual modes an analyst may be in, as well as the actions the system performs in response to an analyst being in a particular state. This has the benefit of providing individuals with adaptive responses, while not requiring direct feedback from the analyst in order to do so (e.g., an analyst specifying how much filtering to perform for a particular contextual mode). However, as a possible future improvement, personalized fine-tuning could be incorporated into the system, but would require additional testing to determine the trade-off of added personalization.

These architectural design considerations are inter-woven into the fabric of the resulting system architecture, which speaks to their interconnectedness. As such, it is important to explore these considerations carefully when designing RECON-like systems, as a change in one location can easily impact other parts of the system. Figure 8 summarizes the design considerations discussed in this section, organized according to the five layers of the RECON architecture and the considerations presented in Section IV. To underscore the uniqueness of what is being proposed in this paper, the following section examines related work concerning architectural design considerations.

VI. Related Work

The preceding sections have identified architectural design considerations for adaptive context-aware systems and their application in the recent RECON implementation for the

	Architectural Design Consideration							
	Human Modelling		*Human-Machine Interaction*			*Human-Machine Synergy*		
RECON Component	*Model Selection [GLM vs. Machine Learning]*	*Model Calibration [Individual vs. Collective]*	*Model Transparency [Black-Box vs. White-Box]*	*User Feedback [Direct vs. Indirect]*	*Contextual Inputs [Explicit vs. Implicit]*	*Decision-Making [Individual vs. Team]*	*Cognitive Assistance [Personalized vs. Generalized]*	*Recommendation [User-Based vs. Item-Based]*
Brain-Computer Interface	Machine Learning	Collective	Black-Box	Indirect	Implicit	N/A	N/A	N/A
Human-Computer Interface	N/A	N/A	N/A	N/A	Explicit and Implicit	Individual and Team	Personalized	N/A
Data	N/A	N/A	Black-Box	N/A	N/A	N/A	N/A	N/A
Case-Based Recommender	N/A	N/A	Black-Box	Direct	Explicit	Individual and Team	Personalized	Item-Based
Context	N/A	N/A	White-Box	N/A	Explicit and Implicit	Individual and Team	Generalized	N/A

Figure 8. Design considerations applied to RECON context management.

intelligence analysis domain. These considerations have been motivated by known HCI dilemmas and the cognitive overload problem faced by analysts [1]. The literature on context-aware systems is vast, as seen in [14], and architectures have been proposed that are similar to RECON.

For example, in [50], the authors propose a multi-module approach for a context-aware system middleware having the following modules: a reasoning engine, learning engine, context predictor, access controller, and context integrator. Likewise, in [51], the authors propose an ontology-based decision-support system for the military domain, comprising multiple agents responsible for decision support, user information, available sensors, information services, and context management. Even though these are similar in that they combine multiple tiers for context management, these approaches do not share the same layers as RECON, nor is their emphasis on reducing information overload. While much attention in the literature has focused on such architectures, relatively little has been devoted to the design considerations guiding the development of these systems [14], which is a core focus of this paper. In this section, related work on architectural design considerations is presented and compared with those relevant to RECON, as have been presented in Section IV.

In [47], twelve HCI dilemmas are discussed in the context of supervisory control. A significant number of these are philosophical in nature, such as who should ultimately be in *control*, the human or the machine, and what is the “right” balance between automation and control. These considerations do not relate directly to the RECON system. However, others are more application-oriented, such as the role that *trust* plays in the system and how much trust should be placed in the results coming from automation. Another is how much *free will and creativity* to allow on the part of the user versus having a system that is completely predictable from the designer’s perspective. These two are directly applicable to RECON in terms of both model transparency (i.e., trust) and cognitive assistance (i.e., the extent of user-involvement in the personalization process).

In [52], four design considerations are proposed that directly support two distinct, but related aspects: i) intelligibility of system behaviour and ii) accountability of human users. These considerations include informing the user about the current *capabilities and understanding* of the contextual system, which is in-line with the role of trust in [47] and the idea of model transparency as proposed in Section IV. System *feedback* is also a key feature outlined in [52] and is meant to inform the user about both the consequences of a particular action prior to its being enacted (feedforward) and notification about what the user has done following the action (confirmation). To this end, the authors propose that *identity and action disclosure* be incorporated into a system as part of an audit trail. This differs from RECON in that system feedback is defined by the user through scenes, which then allows the system to provide alerts that are objective-focused. Lastly, *control* is also emphasized and it is noted that the user should have the ultimate control over any actions he or she may be held accountable for. However, in RECON, because the user is restricted to a limited set of actions, including setting objectives and rating documents, this type of control is not a major consideration.

In [53], the design considerations presented are concerned with providing maximum flexibility to business processes. Key issues include how business processes can be *conceptualized* and applied to process models in general. This is not a major concern for RECON, which has been designed and implemented to support the established intelligence analysis cycle outlined in Section II. Another consideration presented in [53] relates to the *contextual variables* used to capture and assist with the business processes. The authors speak to the relevance and the observability of these contextual variables (e.g., some variables might not be observable and may need to be inputted by the user). In RECON, this notion is related to the balance between explicit and implicit contextual inputs wherein objectives and scenes are set explicitly by the analyst, while user-state classification results implicitly from the BCI assessment. The final issue mentioned in [53] is how business processes can be supported in the face of changes to context.

This *flexibility* also relates to the contextual-input consideration in RECON as recommendations, which support the intelligence analysis business process, automatically adapt to changes in context defined by the analyst through explicit objective and scene definitions.

Other researchers have focused on more singular considerations. For example, in [54], the researcher's major design considerations revolve around *enterprise collaboration* and how trust can be improved to support decision making across the entire enterprise. This effort speaks to collaborative management systems and the relevance of research focusing on networked businesses (or "holons" [46]). The major artifact stemming from this work is a table of trust criteria that can be used when implementing such systems. This relates closely to RECON's consideration of individual versus team decision-making, which is crucial as organizations increasingly must coordinate efforts in order to manage complex situations. Finally, in [55], the major focus area is *privacy* and how it should be managed. The authors propose providing the user with full control over which applications should be given information about the user's present location. While an explicit design consideration was not mentioned, the design considerations implicitly revolved around the problem of privacy and how best to ensure it. In terms of RECON, privacy is not a key consideration, as sharing information is central to sense-making among analysts, and those receiving an analyst's information are considered to be trustworthy. It remains to be fully investigated how much implicit information, like an analyst's brain-state classification from EEG signals, is appropriate to be shared with other members of the organization. Such a policy would necessarily need to be determined on an organizational basis.

Each of the foregoing, while being related to context-aware systems, presents a unique perspective that highlights distinct architectural design considerations. As seems natural, these considerations are heavily motivated by the problem under investigation. For example, in [55] the authors focus on privacy, so the architectural considerations in the paper relate to how best to support user-controlled privacy. For RECON, the uniqueness of the solution, in terms of combining many different, yet relevant and supporting techniques, brings with it a unique set of considerations, which are not always considered in one system. Hence, the architectural design considerations described in Section IV offer a foundation from which future research attempting to create an adaptive, context-aware solution to the problem of cognitive overload in intelligence analysis can begin.

VII. Conclusion and Future Work

With its focus on cognitive offloading and high-relevance system recommendations, RECON targets the adaptive context-aware systems domain for intelligence analysts through a unique five-layer architecture having an explicit human-factors view of context management. Eight key architectural design considerations have been proposed herein for context-aware support, and their application to the implemented RECON system has been presented. Moreover, previous architectural discussions have been extended in this paper with a detailed recommendation algorithm and a cognitive model for context classification.

It is expected that in situations involving information overload, uncertainty, and time pressure, the effectiveness of intelligence analysts can be significantly improved through context-aware adaptive systems, where these design considerations have been conscientiously applied. However, there remains room for more comparisons and discussion of these considerations in light of future system implementations. Also, more practical testing of the architectural implementation is required to ascertain its ability to support analysts. As part of future work, a human-in-the-loop experiment will investigate the effectiveness of the RECON implementation and approach in reducing cognitive overload based on the principles of adaptive-context management.

Acknowledgment

This work was funded by Defence R&D Canada, by Thales Research and Technology Canada, and by a research partnership grant from the Department of National Defence of Canada and the Natural Sciences and Engineering Research Council of Canada. The authors would also like to acknowledge the reviewers, both from this journal and the ADAPTIVE 2014 conference, for their thoughtful feedback.

References

[1] 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 2014, The Sixth International Conference on Adaptive and Self-Adaptive Systems and Applications, 2014, pp. 68–72.

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

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

[4] Central Intelligence Agency, "The work of a nation," Library of Congress, Tech. Rep., 2009.

[5] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," in Proceedings of International Conference on Intelligence Analysis, vol. 5. Mitre McLean, VA, 2005, pp. 1–6.

[6] ——, "Information foraging," Psychological review, vol. 106, no. 4, 1999, p. 643.

[7] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card, "The cost structure of sensemaking," in Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems. ACM, 1993, pp. 269–276.

[8] 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.

[9] E. S. Patterson, D. D. Woods, D. Tinapple, E. M. Roth, J. Finley, G. G. Kuperman, and H. E. Directorate, "Aiding the intelligence analyst in situations of data overload: From problem definition to design concept exploration," Institute for Ergonomics/Cognitive Systems Engineering Laboratory Report, ERGO-CSEL, 2001.

[10] M. Chesbro, "Intel-cyclopedia: A guide to sources of information for the intelligence community," Homeland Security Digital Library, 2011, retrieved: April 2014.

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

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

[13] 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.

[14] 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, 2009, pp. 8509–8522.

[15] D. Gouin, V. Lavigne, and A. Bergeron-Guyard, "Human-computer interaction with an intelligence virtual analyst," in Proceedings of Knowledge Systems for Coalition Operations, IHMC, Pensacola, FL, 2012, pp. 1–5.

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

[17] B. Graimann, B. Allison, and G. Pfurtscheller, "Brain–computer interfaces: A gentle introduction," in Brain-Computer Interfaces. Springer, 2010, pp. 1–27.

[18] ——, Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction. Springer, 2010.

[19] R. Ramirez and Z. Vamvakousis, "Detecting emotion from EEG signals using the emotive epoc device," in Brain Informatics. Springer, 2012, pp. 175–184.

[20] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," Sensors, vol. 12, no. 2, 2012, pp. 1211–1279.

[21] A. Holm, K. Lukander, J. Korpela, M. Sallinen, and K. M. Müller, "Estimating brain load from the EEG," The Scientific World Journal, vol. 9, 2009, pp. 639–651.

[22] D. O. Bos, "EEG-based emotion recognition," The Influence of Visual and Auditory Stimuli, 2006, pp. 1–17.

[23] J. A. Russell, "Core affect and the psychological construction of emotion." Psychological review, vol. 110, no. 1, 2003, p. 145.

[24] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "DEAP: A database for emotion analysis using physiological signals," Affective Computing, IEEE Transactions on, vol. 3, no. 1, 2012, pp. 18–31.

[25] AlchemyAPI Website, URL: http://www.alchemyapi.com (Last accessed: 2014.11.12).

[26] OpenCalais Website, URL: http://www.opencalais.com (Last accessed: 2014.11.12).

[27] W. Ross, M. Ulieru, and A. Gorod, "A multi-paradigm modeling and simulation approach for system of systems engineering: A case study," in IEEE 9th International System of Systems Engineering Conference, 2014, pp. 1–6.

[28] L. Candillier, M. Chevalier, D. Dudognon, and J. Mothe, "Multiple similarities for diversity in recommender systems," International Journal On Advances in Intelligent Systems, vol. 5, no. 3 and 4, 2012, pp. 234–246.

[29] 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, 2002, pp. 339–351.

[30] E. Hollnagel, "Context, cognition, and control," in Co-operative Process Management, Y. Waern, Ed. London: Taylor & Francis, 1998, pp. 27–52.

[31] S. Makeig, C. Kothe, T. Mullen, N. Bigdely-Shamlo, Z. Zhang, and K. Kreutz-Delgado, "Evolving signal processing for brain–computer interfaces," Proceedings of the IEEE, vol. 100, no. Special Centennial Issue, 2012, pp. 1567–1584.

[32] C. Mühl, B. Allison, A. Nijholt, and G. Chanel, "A survey of affective brain computer interfaces: principles, state-of-the-art, and challenges," Brain-Computer Interfaces, no. ahead-of-print, 2014, pp. 1–19.

[33] A. Lécuyer, L. George, and M. Marchal, "Toward adaptive VR simulators combining visual, haptic, and brain-computer interfaces," Computer Graphics and Applications, IEEE, vol. 33, no. 5, 2013, pp. 18–23.

[34] D. Tan and A. Nijholt, "Brain-computer interfaces and human-computer interaction," in Brain-Computer Interfaces. Springer, 2010, pp. 3–19.

[35] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity," User Modeling and User-Adapted Interaction, vol. 24, no. 1-2, 2014, pp. 35–65.

[36] P. G. Campos, I. Fernández-Tobías, I. Cantador, and F. Díez, "Context-aware movie recommendations: An empirical comparison of pre-filtering, post-filtering and contextual modeling approaches," in E-Commerce and Web Technologies. Springer, 2013, pp. 137–149.

[37] T. Tsiligkaridis, B. Sadler, and A. Hero, "A collaborative 20 questions model for target search with human-machine interaction," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, May 2013, pp. 6516–6520.

[38] G. D. Hutcheson and N. Sofroniou, The multivariate social scientist: Introductory statistics using generalized linear models. Sage, 1999.

[39] M. A. Pitt, W. Kim, and I. J. Myung, "Flexibility versus generalizability in model selection," Psychonomic Bulletin & Review, vol. 10, no. 1, 2003, pp. 29–44.

[40] S. Roberts and H. Pashler, "How persuasive is a good fit? a comment on theory testing." Psychological review, vol. 107, no. 2, 2000, pp. 358–367.

[41] W. Estes and W. T. Maddox, "Risks of drawing inferences about cognitive processes from model fits to individual versus average performance," Psychonomic Bulletin & Review, vol. 12, no. 3, 2005, pp. 403–408.

[42] P. N. Mohr and I. E. Nagel, "Variability in brain activity as an individual difference measure in neuroscience?" The Journal of Neuroscience, vol. 30, no. 23, 2010, pp. 7755–7757.

[43] A. Schmidt, M. Kranz, and P. Holleis, "Interacting with the ubiquitous computer: towards embedding interaction," in Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies. ACM, 2005, pp. 147–152.

[44] S. Schiaffino and A. Amandi, "User–interface agent interaction: personalization issues," International Journal of Human-Computer Studies, vol. 60, no. 1, 2004, pp. 129 – 148.

[45] D. S. Alberts and R. E. Hayes, "Power to the edge: Command... control... in the information age," DTIC Document, Tech. Rep., 2003.

[46] W. Ross, A. Morris, and M. Ulieru, "NEXUS: A synergistic human-service ecosystems approach," in Sixth IEEE Int'l Conf. Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2012, pp. 175–180.

[47] T. B. Sheridan, "HCI in supervisory control: Twelve dilemmas," in Human error and system design and management. Springer, 2000, pp. 1–12.

[48] A. Subasi, "EEG signal classification using wavelet feature extraction and a mixture of expert model," Expert Systems with Applications, vol. 32, no. 4, 2007, pp. 1084–1093.

[49] ——, "Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction," Computers in Biology and Medicine, vol. 37, no. 2, 2007, pp. 227–244.

[50] W. Chun-dong, L. Xiao-qin, and W. Huai-bin, "A framework of intelligent agent based middleware for context aware computing," in Natural Computation, 2009. ICNC'09. Fifth International Conference on, vol. 6. IEEE, 2009, pp. 107–110.

[51] 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.

[52] V. Bellotti and K. Edwards, "Intelligibility and accountability: human considerations in context-aware systems," Human–Computer Interaction, vol. 16, no. 2-4, 2001, pp. 193–212.

[53] M. Rosemann and J. C. Recker, "Context-aware process design: Exploring the extrinsic drivers for process flexibility," in The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium. Namur University Press, 2006, pp. 149–158.

[54] P. Kaur, S. Ruohomaa, and L. Kutvonen, "Enabling user involvement in trust decision making for inter-enterprise collaborations," International Journal on Advances in Intelligent Systems, vol. 5, no. 3 and 4, 2012, pp. 533–552.

[55] F. Dorfmeister, S. Feld, and C. Linnhoff-Popien, "ALPACA: A decentralized, privacy-centric and context-aware framework for the dissemination of context information," International Journal On Advances in Intelligent Systems, vol. 7, no. 1 and 2, 2014, pp. 223–236.