



and Societal (IOS) (Zacharias, 2008), Human, Social, Culture Behavior (HSCB) (Hartley, 2008), and Diplomatic, information, military, and economic (DIME), political, military, economic, social, information, and infrastructure (DIME/PMESII) (Zacharias, 2008) to name a few.

Efforts and tools categorized as IOS, HSCB, and DIME/PMESII represent a broad set of capabilities involving a multi-discipline of social and human behavioral sciences for analyzing data and using knowledge bases, theories, models, computational models, and training to understand, predict, and affect cultural behavior (Hartley, 2008; Zacharias, 2008). An organized community of HSCB participants has recently emerged comprised of a mix of operational users, operations researchers, social scientists, and technologists. In a recent HSCB modeling workshop, operational users of this community expressed their needs for HSCB capabilities to predict outcomes for a situation, support mission activities, balance activity flow such as those involved in Security, Stability, Transition, and Reconstruction (SSTR) (West, 2008; DoD Directive 3000.05, 2005), identify priorities, and understand cultures and responses to events (Hartley, 2008).

In general, interoperability is the ability of two or more systems to exchange information and to make use of the exchanged information (IEEE, 1990). There are different levels of interoperability that are generally categorized as syntactic involving communicating and exchanging data between systems and semantic interoperability involving systems that can interpret and process the data that is exchanged (Tolk, 2003; Turnitsa, 2005; Selvage, 2006). Details about interoperability levels are in (Tolk, 2003; Turnitsa, 2005).

For phase 1 and 2, IBAC integrated analytical tools that process volumes of unstructured data originating from text-based documents that may include documents, speeches, video data, news, and web pages from multiple sources. The integration and interoperability problem for IBAC has been to construct a semi-automated semantic model for this data. This is particularly important since separate tools may process the same data but provide different results and introduce different errors due to their different analysis and algorithms.

From lessons learned from prior efforts, the life-cycle effort involved, and the differing versions available (Faatz 2002; Langton, 2007; Hartley, 2008; Zacharias, 2008), the IBAC team did not formulate or incorporate a social science ontology. Rather IBAC provides the user with the capability to form data models from the perspective of composing an assessment. That is, the basis for the data model involves the assessment elements such as the

objective, outline, terms, content, supporting materials and its essential elements of information. IBAC utilizes semantic analysis from tools such as WordNet (Fellbaum, 1998) to assist with engaging the analytical tools and composing assessments. This is in contrast to model-to-model interoperability that involves solving the complexity of incompatibility types as interface, ontological, formalism, and sub-domain gaps (Langton, 2007). IBAC provides the syntactic interoperability at the interface levels following the best practice guidance of (DoD-NCSS, 2007; Holley, 2006; Selvage, 2006; Duan 2007) and addresses the ontological and sub-domain gaps mostly from the perspective of unstructured data and does not address a solution for metamodels (Tolk, 2004), data values and types, and algorithm formalisms as in (Langton, 2007).

In this paper, an overview of the interoperability benefits and challenges is presented that establishes the context for introducing the IBAC open architecture. A technical discussion on the architecture includes the IBAC services, the use of SOA, and the assessment work flow. The IBAC approach to interoperability is described at the user interface (UI) level, called the Common User Interface (CUI), and at the service level using a simple XML messaging schema that addresses syntactic interoperability. Semantic interoperability is described for semantic similarity analysis involved with engaging service-based CBM capabilities and for the software infrastructure to accommodate metadata and ontologies as they become more established by the community.

## 2. The Challenges of Interoperability

Interoperability among CBM tools provides the benefits of leveraging the best-of-breeds in tools, connecting two or more tools ad hoc with little or no manual intervention, semantically comparing results, reducing duplication of capabilities and efforts through re-use, and yielding better economics. The challenges of interoperability are not new as can be seen in the plethora of literature as (DoD-NCDS, 2003; DoD-NCSS, 2007; Tolk, 2003; Duan, 2007; SEI, 2008; Holley, 2006) to identify a few. Challenges to achieving interoperability involve the following (Hartley, 2008; Zacharias, 2008; ONR-BAA, 2008):

- Developing common CBM metadata, taxonomies, and ontologies
- Identifying methods and structures for organizing source data, data from collection systems, and Human Intelligence (HUMINT)
- Utilizing multiple and disparate data sources including open source repositories
- Having a collaborative modeling or service framework, a so-called service bus, that can link

multiple HSCB tools and other services in a federated assembly

- Providing a workstation or digital notebook (ONR-BAA, 2008) for supporting field operators that utilizes CBM tools and support utilities to support a focused region of interest.

### 3. The IBAC Open Architecture

IBAC is a web-based software integration platform that provides services for assessment authoring designed for a plug-and-play service extension for engaging behavior-based modeling and analytic tools such as CBMs and HSCBs. IBAC is intended to be independent of any specific domain space. IBAC becomes specialized through integrated domain-specific and semantic processing services. To accomplish this, the IBAC software architecture is designed for interoperability using Service Oriented Architecture (SOA) design patterns and guidance (Ferguson, 2005; Krafzig, 2005; Duan, 2007), Web 2.0 principles and practices (O'Reilly, 2005), proven software technologies, and the Representational State Transfer (REST) architectural styles (Fielding, 2000). IBAC is designed to be an open system that provides services on-demand and supplies an open system user interface, called the CUI, for its end users. As a system of systems platform, IBAC is also designed using the best practices (SEI, 2008) to evolve with changes in technology, policies, objectives, services, and communications.

IBAC is organized in the following functional areas or categories (Brown, 2008):

- Assessment processing
- Common User Interface (CUI)
- Data Centric Capability - Search and Query
- Theme Analysis
- HSCB Service Integration
- Repository Management
- Publishing

These functional areas were derived from the IBAC Requirements including use cases and the work flow for producing assessments. The IBAC system level view is illustrated in Figure 1.

A software system implemented as SOA is infrastructure comprised of a collection of services that can execute on different computers on the Internet (Ferguson, 2005). These services are orchestrated together in a seemingly ad hoc manner to execute a process flow that corresponds to a use case for a problem domain. For IBAC, the underlying domain is the assessment work flow.

Services orchestrated with IBAC are self-contained entities that can be built to solve a particular problem, provide mission or business value, host some data, and to contain one or more CBMs. IBAC becomes specific to a domain space by adding domain-specific services with corresponding semantic support. Characteristics of a SOA implementation include (Brown and Johnson, 2002; Ferguson, 2005):

- Services are loosely coupled - there is no dependency between each running service,
- Services have well-defined interfaces with the interface defined at the application or domain level (e.g. extract social network, assert belief) rather than at a technical level (e.g. start database transaction),
- Services communicate by synchronous or asynchronous means through some common message format,
- Services can be discovered through a service registry system.

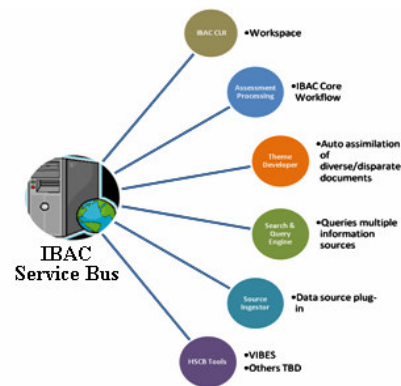


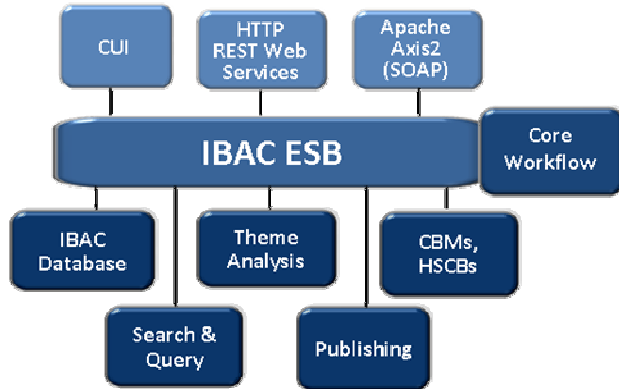
Figure 1 IBAC System Level View

SOA-based systems are typically described using four primary components (Krafzig, 2005):

- Application Front-end – the application user interface that initiates and controls the activity or domain process flow for the enterprise system
- Service – as described earlier a service provides useful domain functionality and is used by application front-ends and other services; also consists of a service contract that specifies the usage of its interface and the interface itself
- Service Repository – A searchable directory that facilitates service registration and contains information for finding and using Services
- Service Bus – primarily used to connect application front-ends and services

Another useful technical description of a SOA system is comprised of two main parts as service endpoints and the message transport fabric called the Enterprise Service Bus

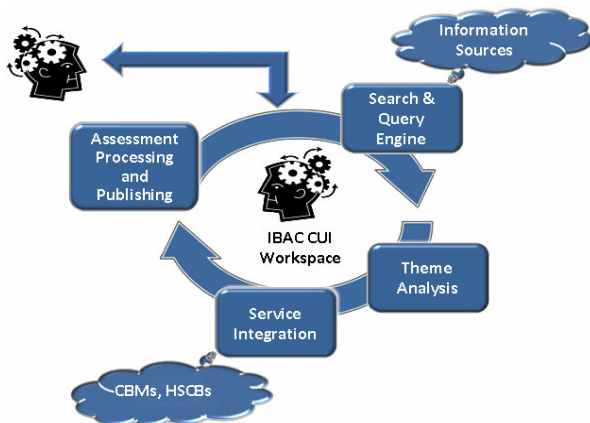
(ESB) that facilitate endpoints to communicate (Ferguson, 2005). The ESB and the Service Bus from (Krafzig, 2005) are the same. Figure 2 illustrates this technical diagram tailored for the IBAC system in a horizontal configuration.



**Figure 2 IBAC Service Architecture Enterprise View in a horizontal configuration**

### 3.1 Assessment Process flow

In general, analysts produce assessments through activities such as searching, information gathering, reviewing information, making notes, assimilating, analyzing, organizing, visualizing, writing, editing, publishing, and iterating as needed. These activities can be performed at any time; however, typically the user follows a sequence and then iterates on that sequence. This sequence, called the Assessment Development Process Flow, or more simply, the assessment work flow, involves performing queries, reviewing and selecting documents and data, making notes, analyzing with tools, and then composing draft entries to the assessment. All of this is done through the CUI. This flow also has a larger scale perspective that identifies the progress state of the assessment activities and development as is illustrated in Figure 3.



**Figure 3 Assessment Development Progress Flow**

## 4. IBAC Interoperability

In its current release, IBAC provides a mechanism for the syntactic part of service interoperability, addresses the semantic component involving searches and engaging service feeds and results, and a form of interoperability at the user interface. Systems that operate with IBAC can exchange messages through simple communications and data formats following the guidelines from REST (Fielding, 2000). For semantic processing, WordNet (Fellbaum 1998) is used for supplementing searches from multiple data sources and disambiguating entries and results from multiple HSCB tools. Interoperability that is often overlooked is at the user interface. At its simplest form, IBAC facilitates typical user interactions for moving data between applications and tools with the added intelligence for tracking and aiding the analyst with those interactions.

### 4.1 Interoperability at the UI Level

The analyst uses the IBAC system through the CUI also referred to as the assessment workspace. The terms CUI and workspace will be used interchangeably. Within the CUI, a rudimentary semantic interoperability is provided when the user manually moves data between applications, e.g. models, analysis tools, and the editor, through familiar UI metaphors (Szabo, 1995) as copy-and-paste and drag-and-drop.

These user interface operations by themselves offer little for addressing the required interoperability. However, IBAC provides value-added capabilities that are triggered by these actions that include automatic document reference association and text classification. When a text is selected, copied, and then pasted from a document (e.g. a web page, an HSCB text result page, or other forms) to the editor, the CUI automatically records this action, formulates a message, and exchanges this message with the IBAC Service Bus on the server-side. The server-side handles this message by identifying the source document as a reference, identifying the selected text, adding the source identifier to the citation list of the assessment, forming and saving a record, and notifying the CUI of these actions. All of this happens unbeknownst to the analyst.

The selected text is further analyzed against the metadata of the assessment. This assessment metadata is organized by the outline of the assessment where each outline topic with its content is analyzed and assigned tags forming a familiar bag-of-words model. When a text is copy-pasted to the editor, it is processed into tags and compared with the tags for each topic yielding a rank order of results

based on matches. This is used as an aid to the analyst for identifying similarities of the selected text across all topics within the assessment. Synsets from WordNet are used to group similar tags within each topic and within the text segment and then used to expand tags in preparation for performing the comparisons. As content is changed or added to the assessment, the metadata is re-evaluated, updated and comparisons are repeated as needed.

With its technology based on Web 2.0, the CUI facilitates users to open multiple application windows within a single browser, share information, add new tools and resources, and compose their assessment with the editor. Essentially the CUI provides an on-line version of a Personal Computer (PC) desktop in a browser page. The benefit of this is that the CUI is a web-based virtual desktop where analysts can save and recall their settings such as application windows from any computer that supports a web browser. The user's workspace and his assessment content are stored on the server-side on one or more servers following the Cloud Computing style (Chappell, 2008).

The workspace provides access to online tools for authoring, searching, document analysis, behavioral analysis tools (e.g. an HSCB tool) and other services (e.g. a text analysis visualizer) as driven by the assessment workflow described earlier. The workspace also provides the scope or context for the assessment and its related subject matter. In this capacity, the workspace is a container for the assessment, its drafts, supporting reference documents, the application windows, analysis results, notes, annotations, searches and search results, and a history of actions and events that have been performed by the analyst within the workspace.

Some of the CUI capabilities that involve interoperable services from the user perspective include:

- Automatically identifying text segment matches to assessment topics,
- Connecting to HSCB web-enabled tools and using their native HSCB UI for performing analysis,
- Pushing data to one or more HSCB web-enabled tools,
- Adding a service, for example an HSCB tool, in an ad hoc manner,
- Using familiar and easy-to-use user interface actions such as copy-paste and drag-and-drop for doing complicated 'back-end' actions such as passing data to an HSCB tool for processing.

Figure 4 provides a screen shot for an IBAC CUI workspace illustrating the assessment editor on the far right with some content, the workspace navigation

controls as shown horizontally at the top, the workspace tracker on the far left, and a web page, in the middle of the screen, that was selected as a reference for the assessment.

The CUI is a "Rich Internet Application" implemented through Asynchronous JavaScript and XML (AJAX) (Garret, 2005). AJAX is considered a key component of Web 2.0 applications and lends itself to innovations in web-based user interface experiences that are as rich as PC-based applications (O'Reilly, 2005).

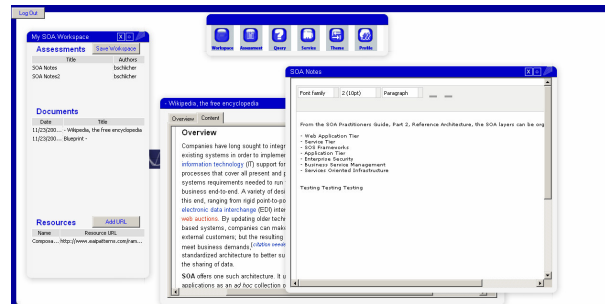


Figure 4 Screen shot of an example IBAC Workspace

## 4.2 Interoperability with Services

To interoperate with IBAC at the syntactic level, an external service communicates using the IBAC Document and ServiceRun XML schemas in addition to implementing the low level features following the REST guidelines. This involves using the familiar and ubiquitous HTTP, URL, resource representations as XML/HTML/images, and MIME types including text/xml, text/html, and the remaining types (Fielding, 2000). This applies to both the IBAC client- and server-side. IBAC data exchange formats are implemented in XML and an additional client-side JSON (JavaScript Object Notation). These were designed to achieve the most basic level of syntactic interoperability.

### 4.2.1 IBAC XML Schema

The principle artifact for IBAC XML is modeled after a document metaphor. The main elements for a document are the title, author, dates, synopsis, and the document content that can be in any encoded format to include any type of data, e.g. text, image. This document-based model was chosen because the early efforts for the IBAC system involved performing text analysis and processing on a corpus of documents. Hence, the document became the unit for communications exchange and additionally a convenience for saving to and retrieving from the IBAC assessment database. This document model is also preserved because of its simple scheme. This schema for this document-based XML is in Figure 5 with the main element called 'artifact'. This name was chosen to avoid

potential confusion with the term ‘document’ that is used for describing XML and used in programming languages for processing XML.

The IBAC Document XML is embedded as one or many in an IBAC message-based XML such as the Service Message Exchange (smsg) that contains a ServiceRun. The ServiceRun XML is a simple package containing an instruction and a collection of documents.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Artifact">
    <xs:complexType mixed="true">
      <xs:sequence>
        <xs:element name="Title" type="xs:string" />
        <xs:element name="Title2" type="xs:string" />
        <xs:element name="ArtifactUID" type="xs:string" />
        <xs:element name="ArtifactURL" type="xs:string" />
        <xs:element name="DataSource" type="xs:string" />
        <xs:element name="Author" type="xs:string" />
        <xs:element name="Synopsis" type="xs:string" />
        <xs:element name="ArtifactDate" type="xs:unsignedLong" />
        <xs:element name="LastUpdate" type="xs:unsignedLong" />
        <xs:element name="CreateDate" type="xs:unsignedLong" />
        <xs:element name="ContentType" type="xs:string" />
        <xs:element name="Content" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

**Figure 5 IBAC Document-based XML schema**

The instructions are designed to be simple for engaging document-based tools and include the following:

```
load_doc, load_doc_execute, load_doc_execute_async,
remove_doc, remove_doc_all, remove_doc_execute,
remove_doc_execute_async, execute, and
execute_async.
```

The names are mostly self-explanatory with the ‘\_async’ term indicating an asynchronous service interaction.

#### 4.2.2 Semantic Support for Service Interoperability

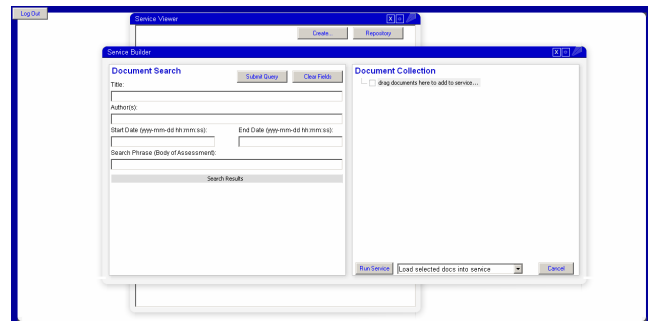
The document content in the artifact XML is either unstructured or structured depending on the interface requirements for the service. The initial analysis tools engaged with IBAC have been based on processing a corpus of unstructured text documents. For these types of tools, IBAC provides a Corpus Manager in the workspace that facilitates the analyst to group, add, remove, and pre-process documents for engaging one or more CBMs for analysis. Since documents may originate with different data formats (e.g. HTML), the text is cleansed of these format characters. The documents are optionally pre-processed for semantic similarity using WordNet (Fellbaum, 1998) for an automated word sense disambiguation (Li, 1995). WordNet was chosen because of its rich source of lexical knowledge, its successful uses, its strong semantic capability for unstructured CBM text processing, and it can be extended to accommodate unique domain knowledge. This semantic analysis can

be similarly applied to the post-processing results from CBMs.

CBMs that do not operate based on the document metaphor are facilitated with an adapter that converts data to and from IBAC and the CBM. Each adapter is custom written to uniquely perform the conversion. This integration approach has been similarly applied for IBAC search services and is in the early stages of development with an agent-based framework.

#### 4.2.3 Interoperability in Use

A simple scenario for invoking a service begins with an IBAC user selecting the ServiceRun window in the CUI as illustrated in Figure 6.



**Figure 6 The IBAC CUI ServiceRun window**

The Visualization of Belief Systems (VIBES) is an Army sponsored product that utilizes state-of-the-art, cognitive-based models and tools (Carley, 2004) for identifying and visualizing beliefs concerning an organization or social group. To arrive at the results, VIBES extracts and semi-automatically determines key actors, resources, and an underlying social network for supporting the belief processing. From its PC-based software platform, VIBES was recently enhanced as a web service with a web browser user interface. In this enhancement, VIBES implemented the IBAC ServiceRun XML and after only three short technical conversations between IBAC and VIBES developers, IBAC was able to successfully push a corpus, start the services, and view the resulting belief analysis from VIBES.

Currently, IBAC interoperates with the Theme Developer service (Brown, 2008) to identify common themes, concepts and relationships within a corpus. Theme Developer provides several views of these results. In an initial set of tests, both VIBES and Theme Developer were used to analyze the same small corpus consisting of 25 related documents. Manual inspect of the resulting visualizations revealed some commonality between the tools with identifying concepts in the corpus. The semantic pre-processing was enabled. A controlled

experiment is underway to determine the semantic results between the tools and to further investigate the IBAC semantic interoperability approach.

## 5. Conclusions

In this paper, an overview of CBM interoperability benefits and challenges was presented that established the context and discussion for the IBAC open architecture to address the challenges with the goal for producing assessments. The software approach for IBAC is based on SOA, with Web 2.0 practices, REST, and Cloud Computing styles. This architecture addresses syntactic interoperability for CBMs by facilitating services and users to successfully exchange information through the IBAC XML, a simple model based on a document metaphor. IBAC provides infrastructure that positions it to readily incorporate CBMs in a federated run-time environment as demonstrated with the integration of VIBES. With the IBAC CUI, an analyst can perform simple manual interoperability functions by moving data between models using familiar UI actions that trigger supporting semantic processing. IBAC is evolving to address the more difficult semantic interoperability issues among CBMs as taxonomies, metadata, and ontologies begin to be more established.

## 6. Future Work

The semantic processing involved with pre- and post-processing for engaging a service will be evaluated for a common corpus of documents in a structured set of experiments. The semi-automated data model constructed with text analysis tools can be further developed to form semantic networks that may include one or more ontologies. The editor technology will be improved that will be able to construct full mashup web pages and offer more possibilities for interoperability through UI metaphors including a cognitive-oriented matching or mirror component.

## 7. Acknowledgements

This material is based upon work supported by the Office of the Secretary of Defense (OSD) with sponsorship from the Defense Threat Reduction Agency (DTRA). We would like to thank Dr. Alenka Brown from the National Defense University (NDU), Matt Holm from DTRA, Joseph Trien from Oak Ridge National Laboratory (ORNL), and Rich Curtis from the Applied Research Laboratory (ARL) at Penn State University.

## 8. References

- Blascovich, J.J., Hartel, C.R., Editors (2008). *Human Behavior in Military Contexts*, National Research Council of the National Academies, The National Academies Press, Washington, D.C.
- Brown, A. (2008). Integrated Behavioral Assessment Capability (IBAC), White Paper, version 2, sponsored by DTRA, September 30, 2008.
- Brown, A., Johnson, S., Kelly, K. (2002). Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications, IBM Corporation. Retrieved from the web Dec. 17, 2008.  
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/2169.pdf>
- Carley, K., Reminga, J. (2004). ORA: Organization Risk Analyzer. Carnegie Mellon University, School of Computer Science, Institute for Software Research International, Technical Report CMU-ISRI-04-106.
- Chappell, D. (2008). A Short Introduction to Cloud Platforms, An Enterprise-Oriented View. Chappell & Associates, sponsored by Microsoft Corporation. Retrieved from the web January 5, 2009.  
<http://www.davidchappell.com/CloudPlatforms--Chappell.pdf>.
- DoD Directive 3000.05. (2005). DoD Directive 3000.05, "Military Support for Stability, Security, Transition, and Reconstruction (SSTR) Operations", November 28, 2005.
- DoD-NCDS. (2003). Department of Defense Net-Centric Data Strategy, May 9, 2003.
- DoD-NCSS. (2007). Department of Defense Net-Centric Services Strategy, May 4, 2007.
- Duan, N. (2007). Towards a Federated SOA Model in Achieving Data Interoperability in DoD. ManTech-MBI. Retrieved from the web December 12, 2008.  
[http://c4i.gmu.edu/events/reviews/2008/papers/22\)Duan.pdf](http://c4i.gmu.edu/events/reviews/2008/papers/22)Duan.pdf)
- Fatz, A., Steinmetz, R. (2002). Ontology enrichment with texts from the www. 6th European Conference on Principles and Practice of Knowledge Discovery in Databases. Retrieved from the web. Feb 14, 2008.  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.901>
- Fellbaum, C., editor (1998). WordNet: An Electronic Lexical Database, Bradford Books.
- Ferguson, D.F., Stockton, M.L. (2005). Service-oriented architecture: Programming model and product architecture, IBM Systems Journal, Vol. 44, No. 4, pp 753-780.
- Fielding, R.T. (2000). Architectural Styles and the Design of Network-based Software Architectures, Dissertation, University of California, Irvine, 2000, Chapter 5. Retrieved from the web Dec. 11, 2008.  
<http://www.ics.uci.edu/~fielding/pubs/dissertation/to>  
p.htm

- Garrett, J.J. (2005) "Ajax: A New Approach to Web Applications". Retrieved from the web Jan. 5, 2009. <http://www.adaptivpath.com/ideas/essays/archives/000385.php>
- Hartley, D. (2008). Human, Social, Cultural Behavior (HSCB) Modeling Workshop I: Characterizing the Capability needs for HSCB Modeling – Final Report", July 2008.
- Holley, K., Palistrant, J., Graham, S. (2006). Effective SOA Governance. IBM Corporation, Retrieved from the web Dec. 17, 2008: <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/soagov-mgmt.pdf>
- IEEE. (1990). Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY.
- IBM Corp. (2006). The Challenge of Securing SOA, IBM Corporation. Retrieved from the web December 19, 2008. [http://www-01.ibm.com/software/sw-library/en\\_US/detail/J120451P64567Q05.html](http://www-01.ibm.com/software/sw-library/en_US/detail/J120451P64567Q05.html)
- Krafzig, D., Banke, K., Slama, D. (2005). Enterprise SOA: Service-Oriented Architecture Best Practices, Prentice Hall, 2004.
- Langton, J., Das, S. (2007). A Framework for Building and Reasoning with Adaptive and Interoperable PMSEII Models. (Final Report #RI-RS-TR-2007-241) Cambridge, MA: Charles River Analytics.
- Li, X., Szapkowicz, S., Matwin, S. (1995). A WordNet-based Algorithm for Word Sense Disambiguation. Proceedings of the 14th International Joint Conference on Artificial Intelligence. pp. 1368-1374.
- ONR BAA 08-025. (2008). Human, Social, Cultural and Behavioral Sciences Broad Area Announcement, Office of Naval Research Expeditionary Maneuver Warfare and Combating Terrorism S&T Department (ONR 30), September 2008.
- O'Reilly, T. (2005). What is Web 2.0 Design Patterns and Business Models for Next Generation of Software. O'Reilly Media, Inc. Retrieved from the web Dec. 10, 2008. <http://www.oreilly.com>
- Software Engineering Institute (SEI). 2008. "Defining Architectures for Interoperability", CarnegieMellon University. Retrieved from the web Dec. 10, 2008. <http://www.sei.cmu.edu/isis/guide/engineering/architectures.htm>
- Selvage, M., Wolfson, D., Bob Zurek, Kahan, E. (2006). Achieve semantic interoperability in a SOA - Patterns and best practices, IBM developerWorks, 23 Jun 2006. Retrieved from the web Dec. 10, 2008. <http://www.ibm.com/developerworks/webservices/library/ws-soa-seminterop.html>
- Szabo, K. (1995). Metaphors and the User Interface. Retrieved from the web December 17, 2008. <http://www.katalinszabo.com/metaphor.htm>
- Tolk, A., (2004). Metamodels and Mappings - Ending the Interoperability War. Proceedings of 2004 Fall Simulation Interoperability Workshop (SIW).
- Tolk, A., Muguira, J.A. (2003). The Levels of Conceptual Interoperability Model (LCIM). Proceedings IEEE Fall Simulation Interoperability Workshop.
- Turnitsa, C.D. (2005) Extending the Levels of Conceptual Interoperability Model. Proceedings IEEE Summer Computer Simulation Conference, IEEE CS Press.
- West, P.D. (2008) "Human, Social, and Cultural Behavior Modeling for Stability, Security, Transition, and Reconstruction Operations, Human, Social, and Cultural", Behavior Modeling Workshop, National Defense University, Washington DC.
- Zacharias, G.L., MacMillan, J., Van Hemel, S.B. Editors. (2008). Behavioral Modeling and Simulation – From Individuals to Societies. The National Academies Press, Washington DC.

## Author Biographies

**BOB SCHLICHER** has over 20 years experience with specifying, designing, building and delivering enterprise software applications and services. As a principal with EigenSoft, he has worked on commercial, military, and government projects involving domain-specific business solutions, analytics, C4I, battle management, knowledge-based systems, service frameworks, interoperability, and web technologies. In his capacity as the Technical Director for the Human Behavior – Information Operations Center (HB-IOC) in the Computational Sciences and Engineering Division at Oak Ridge National Laboratory (ORNL), Mr. Schlicher is responsible for research and development, visualizations, knowledge-based algorithms and analytics, service integration, and deployment of Computational Behavioral Models.

**TRACY WARREN** is Co-Director of the Human Behavior Information Operation Center and an S&T Engineer-Biomedical and Human Factors in the Cyberspace Sciences and Information Intelligence Research Group at Oak Ridge National Laboratory. Her research areas include developing neurocognitive models for the study of human systems including social relationship dynamics, human social learning, and cultural transmission. Ms. Warren has a Master of Science degree in Biomedical Engineering and a Master of Science degree in Experimental Biopsychology, both from Virginia Commonwealth University, Richmond, Virginia.