

Self-Organizing Integration of Competing Reasoners for Information Matching

Sven Brueckner, Elizabeth Downs, Rainer Hilscher, Andrew Yinger

NewVectors, a Division of TechTeam Government Solutions, Inc, Ann Arbor, MI, USA.

{sven.brueckner, liz.downs, rainer.hilscher, andrew.yinger}@newvectors.net

Abstract

Self-organizing systems are robust, scalable, adaptive to a changing environment, and tolerant to noise and incomplete or conflicting information. These are the requirements for our Information Matching System (IMS) that organizes models of document contents and user interest in an abstract information space by relevance to provide any-time recommendations of other users (for collaboration) or documents (for information gathering) to intelligence analysts. In this report on research-in-progress, we present a plug-and-play integration architecture for multiple and possibly competing modelers of arbitrary (text, audio, video, etc) document contents that influence the emerging arrangement of document and user models. The contributions of these modelers are numerical similarity statements that specify attractive or repulsive forces, which guide the ongoing rearrangement of the current set of models. This self-organizing force-based arrangement process adjusts dynamically to changes in the document set or shifting user interest. Our paper also discusses related research, initial experiments that indicate satisfactory system-level behavior, and an upcoming evaluation exercise with actual users.

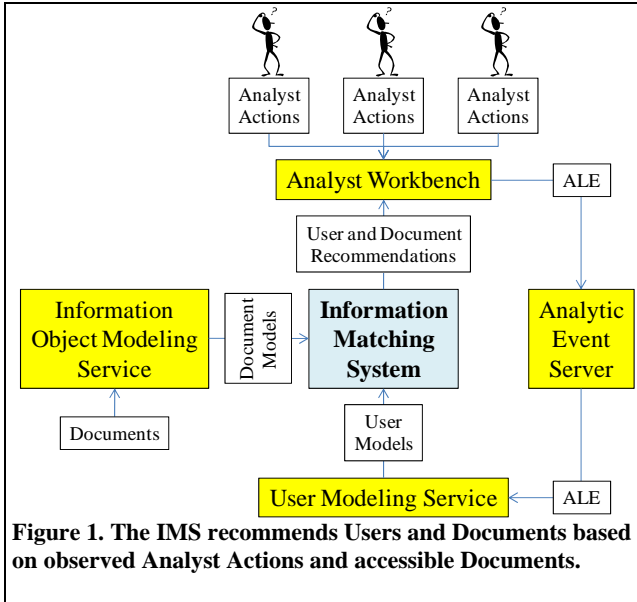
1. Introduction

Our *research-in-progress* reported here in this paper attempts to recommend documents to users (for retrieval) and users to users (for collaboration) in a dynamic environment where the interest of the user and the corpus of available documents is continually changing. To complicate the scenario even further, documents are not necessarily limited to natural multi-language text, but may also include other formats such as audio or video clips and the contents of such documents is hard to interpret unambiguously with machine reasoning. Focusing on the problem of matching users and documents dynamically, we address these challenging requirements by assuming that we have access to a number of potentially competing “Object Modelers” that use their specific approach (e.g., enti-

ty extraction and characterization, keyword frequency analysis, topic modeling, tag correlation, etc) to generate an “Object Model” of a particular document from its raw content. Furthermore, we assume that there is a User Modeling Service (UMS) that dynamically updates a model of the interest of a particular user based on observed user actions (e.g., document access, keyword search, copy or paste of contents) and the object model of any contents associated with such an action. Under these assumptions, we design a self-organizing Information Matching System (IMS) that integrates in its open architecture an arbitrary number of Object Modelers and the UMS to dynamically organize document and user models in an abstract information space where the emergent arrangement of these entities reflects their relevance to each other. The remainder of this paper is structured as follows. In the second section, we describe the challenge problem in a specific application domain. In the third section, we present the architecture and specific algorithms of the IMS. In section four we review some related research, and in section five we show some initial indicators of reasonable system dynamics. Finally, we discuss the upcoming performance evaluation of our prototype in section six.

2. Active Information for Collaborative Intelligence Analysis

We encounter the problem of dynamic information matching in a research program of the US Intelligence Community that seeks to increase the effectiveness of analysts in their execution of particular taskings that require the collection and synthesis of information from the rapidly growing body of intelligence data. The tasking requires an analyst to write a report on a particu-



lar topic, answering a set of specific questions. Thus, the general interest of the analyst is determined by the tasking, but the specific interest changes over time as the analyst focuses on specific aspects of the tasking or explores hypotheses.

As we are supporting a specialized user group (intelligence analysts), we can assume that we can observe their ongoing activities. This assumption does not hold of course for a general user base. In fact, as part of this and previous research programs, logging mechanisms and formats have been developed and integrated with an analyst workbench (nSpace, [11]) that give us access to the stream of “Analyst Logging Events” (ALE) – actions taken by analysts and their associated contents. These ALE’s are consumed by the User Modeling Service (UMS, [2]), which then updates its model of the current interest of the user.

The Information Object Modeling Service (IOMS) integrates a collection of Object Modelers that translate the raw contents of documents into Object Models using specific techniques. Depending on the setup of the system, there may be one or more modelers active in the IOMS and the Document Model presented by the IOMS contains the collection of all Object Models for a given document. By applying different modeling techniques in parallel, the sys-

tem is able to overcome the bias in the interpretation of the document contents (e.g., entities, keywords, topics) that a single approach would introduce.

Figure 1 shows the high-level architecture in which the Information Matching System operates. Its inputs are user and document models from the UMS and IOMS respectively and its output are user and document recommendations for a given analyst. It is important to note here that a user model, just like a document model also comprises multiple Object Models since the contents associated with an ALE (e.g., document viewed, contents copied) is also processed by the IOMS (not shown in Figure 1).

3. Self-Organizing Information Matching

Our Information Matching System (IMS) continuously (re-)organizes a dynamically changing set of user and document models in an abstract information space such that models that are similar to each other remain close in that space while those models that are not similar tend to be farther away.

3.1. What is Self-Organized?

For the purpose of the matching process we unify user and document models (both comprise a set of object models from the IOMS) under the term “InfoPack”. Architecturally (Figure 2), an InfoPack consists of a set of meta-level attributes that describe, for instance, its origin (a particular user ID or the URL of a document) or the level of attention that the system should give it. The InfoPack also carries a list of object model instances, each of which carries model data and an identifier of the specific object modeler type that generated it.

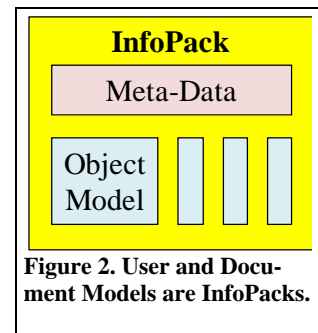


Figure 2. User and Document Models are InfoPacks.

Any given InfoPack relates to a specific entity outside of the IMS. Each user model InfoPack corresponds to a particular analyst that has been or is currently working in the Analyst Workbench (Figure 1). Each document model InfoPack contains the object models for a particular document accessible in the corpus. When the contents of a document changes (as it is for instance the case with Wiki pages) or as the user interest evolves, the object models in the corresponding InfoPacks are updated. If new documents or users “arrive” new InfoPacks are created for them.

3.2. Where does Self-Organization Happen?

Each InfoPack has a unique location in the IMS’ information space that may change over time. The information space is the surface of a torus (Figure 3) that we “create” by either wrapping the edges of a 1x1 square (2D coordinates) or the sides of a 1x1x1 box (3D coordinates). Our initial prototypes used a 2D torus, but recently we decided to move to 3D to provide the InfoPacks with a larger degree of freedom in their movement. The torus surface is necessary to avoid edge effects that would otherwise distort the relative distances between InfoPacks.

The absolute distance between InfoPacks is defined as the distance on the torus. Since our 2D or 3D torus is created by wrapping a square or a box, we can easily calculate the surface distance from the minimum of all distances between one InfoPack’s location (Blue in Figure 4) and the enumeration of all wrapped locations of the other InfoPack (Red in Figure 4).

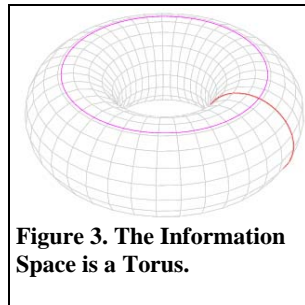


Figure 3. The Information Space is a Torus.

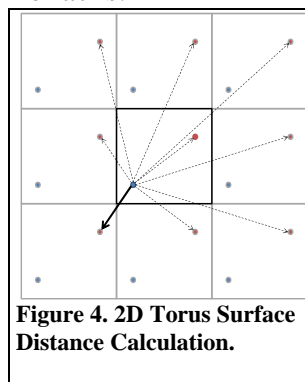


Figure 4. 2D Torus Surface Distance Calculation.

3.3. How does Self-Organization Happen?

In our Information Matching System, the InfoPacks are agents that move in information space (torus), driven by local and non-local interactions with other InfoPacks.

The desired emerging system-level behavior of this agent system is stated as follows:

- Arrange InfoPacks on the torus according to their perceived overall similarity.
- Similarity of two InfoPacks is defined by the object models that they carry. If two object models of the same type are similar in an appropriate similarity metric, then this measure should contribute to the overall similarity of the InfoPacks.

From this global goal, we derive the following agent “intentions”:

- 1) InfoPacks that are not known to be similar do not “want to” be close in information space.
- 2) InfoPacks that are known to be similar “want to” be close in information space.

Then we translate these intentions into notional forces that drive each InfoPack’s movement from its current location in information space:

- 1) Repulsive Force (local): An InfoPack in information space is pushed away from neighboring InfoPacks with a strength decreasing by distance.
- 2) Attractive Force (non-local): An InfoPack in information space is pulled towards any InfoPacks that are known to be similar.

On the one hand, the repulsive force on an InfoPack’s location is based directly on the distances on the torus (local interaction). The closer two InfoPacks are on the torus, the stronger (longer force component vector directed away from the repelling InfoPack) should this force be. For computational simplicity, we cut off this force at a certain distance on the torus. Without the presence of any attracting force components the indiscriminate repulsive forces result in an arbitrary homogeneous arrangement of all InfoPacks across the torus.

On the other hand, the strength of the attractive forces on an InfoPack's location is not determined by the distance relationships on the torus (non-local interaction). Rather, their strength is modulated by the similarity of InfoPacks as perceived from the perspective of a particular object model type carried by the InfoPacks. Thus, InfoPacks may be attracted to any other InfoPack on the torus regardless of their location, as long as these two InfoPacks carry similar object models.

Any InfoPack will apply attractive force components (towards other InfoPacks) if these InfoPacks are similar in any of their object models. If the InfoPacks are similar across multiple object model types, then more than one attractive force component is

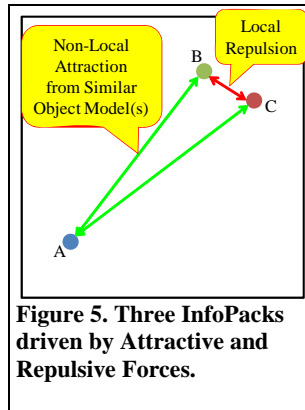


Figure 5. Three InfoPacks driven by Attractive and Repulsive Forces.

calculated. The strength of the force (length of the component vector towards the other InfoPack's location in the torus) is proportional to their similarity for a particular object model type. Thus the more similar the object models are, the stronger the attractive force will be. Figure 5 illustrates these forces in a 3-InfoPack example.

An InfoPack will calculate all attractive and repulsive force components as a function of its current location on the torus and the object-model similarities of all other InfoPacks. Once the weighted sum (weights controlling the importance of the two counteracting intentions) of all these attractive and repulsive force components is computed, the InfoPack will move a length-limited step into the direction of this combined force.

This movement process is repeated for all InfoPacks in the IMS indefinitely, but it approaches relative convergence as the pattern of attractive and

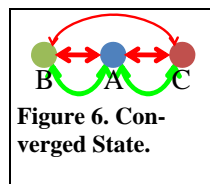


Figure 6. Converged State.

repulsive forces balances out in a minimum energy state. Figure 6 shows a possible energy-minimizing state for the InfoPacks in Figure 5.

3.4. How is it Implemented in Detail?

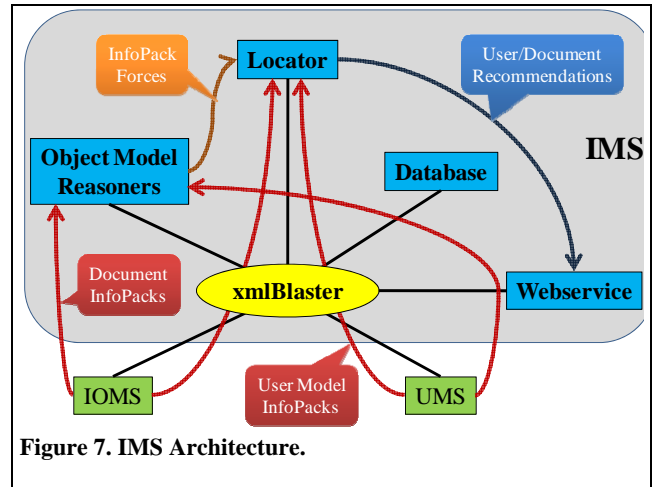


Figure 7. IMS Architecture.

We designed the architecture of the Information Matching System to be flexible and scalable. Figure 7 shows the main component of the IMS:

- A collection of **Object Model Reasoners**, each assigned to a particular object model type active in the IOMS.
- The **Locator**, which maintains the information space (torus) with InfoPacks located in it.
- A **Database**, which maintains a persistent image of all InfoPacks in their current state.
- A **Webservice**, which provides the recommendation service to the external Analyst Workbench.

These components, as well as the IOMS and the UMS, are all connected to an xmlBlaster server [1], which allows us to distribute them across the network on different hosts to maximize the utilization of processing cycles. Through xmlBlaster publish/subscribe protocols, document and user model InfoPacks are submitted by the IOMS and the UMS to all Reasoners, the Locator, and the Database. Similarly, the Reasoners announce specific forces (see section 3.3) between pairs of InfoPacks to the Locator, which hosts the agents that move the InfoPacks

across the torus. Finally, the Webservice component (currently run in Apache Tomcat 5.5) contacts the Locator for lists of user or document neighbors on the torus for a specified user model InfoPack.

Specific Force Calculations

The specific logic of an Object Model Reasoner depends on the type of model on which it bases its InfoPack comparison on. In the following, we provide two examples from our current implementation.

SCM Reasoner.—Specialized Concept Maps [6] are graphs of named entity nodes where each entity is of a certain specialization (e.g., Person entity for social networks, Place entity for spatial networks) and where the relationships (in our current implementation) denote the co-occurrence of these entities in a sentence or paragraph in a given text document from which the map was derived. Furthermore, each such entity carries selected attributes that are derived from external databases (e.g., age/gender for Person, lat/lon for Place).

A simplistic SCM reasoner would compare the SCM's carried by two InfoPacks and compute

their similarity as the average distance between entities of the same specialization based on their attribute values in a normalized attribute space (Figure 8). Thus the similarity value is limited to the [0,1] interval, which determines the strength of the attractive forces between these two InfoPacks communicated to the Locator whenever InfoPacks are added or their SCM models are updated. Alternatively, the Reasoner could ignore the entity attributes and deter-

mine SCM model similarity from the overlap of the named entities carried by the two InfoPacks.

Topic Model Reasoner.—There is a whole class of content modeling techniques that uses statistical methods to determine a set of topics that describe a large corpus of documents, where a topic is a probability distribution over a set of relevant keywords. A topic model of a particular document (not necessarily from that corpus) is then a probability distribution over these topics that determine the likelihood that a particular topic is representative of the document contents.

A Topic Model Reasoner measures the similarity of the topic probability distributions for two InfoPacks. We implemented the Hellinger Distance metric (see [7] for introduction), a classical measure of the similarity of discrete probability distributions. This similarity determines the strength of the pair-wise InfoPack forces communicated to the Locator.

Force-Based Arrangement of InfoPacks

The various Object Model Reasoners in the IMS communicate forces based on pair-wise InfoPack similarity for a particular object model

type to the Locator component. A force is described by four attributes: the object model type, the id's of the two InfoPacks, and the strength of the force (between -1 and +1). The Locator updates the list of known forces for all InfoPacks in the system. The Locator is the execution environment for the InfoPack agents that move their InfoPacks on the surface of the torus that makes up the information space. It creates new agents when new InfoPacks

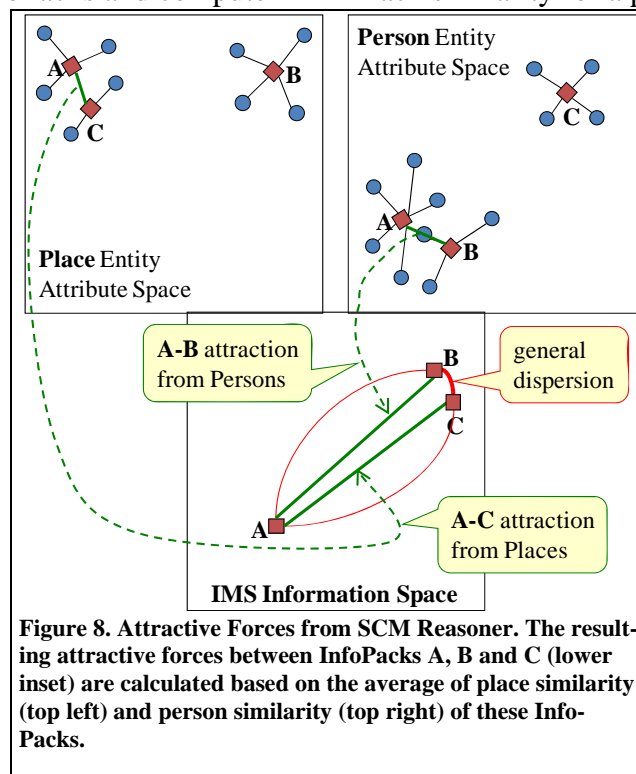


Figure 8. Attractive Forces from SCM Reasoner. The resulting attractive forces between InfoPacks A, B and C (lower inset) are calculated based on the average of place similarity (top left) and person similarity (top right) of these InfoPacks.

arrive or removes them when the InfoPack is deleted, and it continuously iterates over the list of agents to give each a chance to update its location based on the current force pattern.

An InfoPack agent, when activated by the Locator, iterates over the list of forces that include its InfoPack's id. For each InfoPack that is also part of such a force, it determines the shortest vector on the torus to the current location of the InfoPack. This vector provides the direction of the force, but the strength of the force and whether it is attractive (positive strength) or repulsive (negative strength) is set by the reasoner that communicated this force. The agent scales the vector accordingly and averages all these force-component vectors. *The resulting vector represents the combined guidance by all reasoners.*

In a second step, the agent iterates over all InfoPacks in its immediate neighborhood (up to a given threshold) on the torus. For all these neighbors it averages repulsive component force vectors that point away from the InfoPacks and whose length is inversely proportional to their distance to the agent's InfoPack. *The resulting vector represents the general uncertainty or lack of knowledge of the relationships between InfoPacks.*

Finally, the agent adds the guidance vector from the reasoners and the uncertainty vector from its neighborhood to determine the direction of its next step on the torus. If the length of this vector is below a globally defined step-length threshold, then the next step will be limited in length to allow the InfoPacks to settle down in equilibrium. Otherwise, the agent will take a step equal to this threshold in the direction of this vector. Limiting the length of the agent's move allows the system as a whole to gradually evolve towards equilibrium on complex individual trajectories, avoiding thrashing and other pathological emergent patterns.

4. Related Research

The Force-Based Arrangement technique for the self-organization of InfoPacks that we apply in

the Information Matching System is related to iterative embedding techniques traditionally used in graph-drawing algorithms (e.g., Force-Directed Placement [5], "spring embedding" used in Mathematica [12]). These techniques are a form of Multi-Dimensional Scaling (MDS, see [4] for instance) for dimensionality reduction of complex data sets.

We extend these techniques in two novel ways. First, we continue to execute our agents even when they reach equilibrium to account for changes in the dataset over time – as the collection of InfoPacks changes, the system will re-organize to reflect the changing InfoPack similarities. Secondly, we integrate multiple methods (object modelers and reasoners) to determine InfoPack similarity, allowing for potentially conflicting assessments which we balance out as we reach the next equilibrium.

Both these extensions set our approach apart from [9] where force-based self-organization of a multiagent system is used to solve the facility positioning problem. [9] applies force-based self-organization to optimize a given set facilities with respect to distance. Our system does not optimize a specified fitness function and it has to deal with a continuous influx of new InfoPacks. Also, [9] only employs a single "reasoning" mechanism. The Force-Based Arrangement technique is also related to classical field-based stigmergic coordination techniques that emulate pheromone dynamics [3] or use the co-fields approach [8]. But rather than maintaining an explicit field in the environment that guides the movement of agents through the local gradient and the response of the particular agent personality to the vocabulary of fields, we allow for immediate long-range interactions through the iterative combination of component forces. We already demonstrated the use of this technique in guiding robotic vehicles to dynamically arrange around multiple targets in physical spaces [10]. Now the Information Matching System expands it to abstract information spaces.

5. Initial Experiments

Other project and customer priorities have kept us thus far from performing a systematic evaluation of the performance of the Information Matching System, but we can report some initial indicators that show that the self-organization of InfoPacks driven by Reasoner forces results in the desired emergence of clusters of InfoPacks with high similarity in the information space.

5.1. Absence of Guidance Experiment

We claim in section 3.3 that absent any guidance forces from the Reasoners, the localized repulsive uncertainty forces will result in an arbitrary but approximately homogeneous distribution of InfoPacks across the torus. Experiments with 40 InfoPacks without any active reasoners indeed show no distinguished clusters.

5.2. Large Artificial Dataset Experiment

In another experiment we created 1000 artificial InfoPacks with only an SCM object model where each InfoPack's model carried only one Person and one Place entity. The models for each InfoPack were created such that the dataset contained four clusters (with some spread) of 200 InfoPacks each. The remaining 200 InfoPacks were created randomly across the entire data range. Figure 9 shows the "ground truth" in the data, where the black dots are the single Place entities in the SCM models of the InfoPacks and the green lines are connecting entities that are considered close enough by the SCM Reasoner to communicate attractive forces to the Locator.

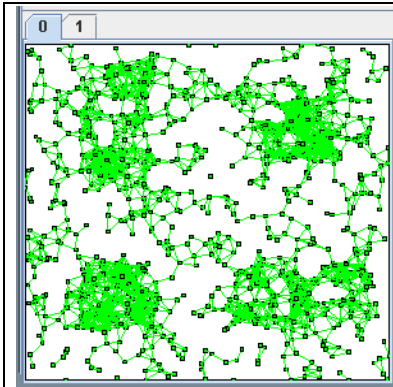


Figure 9. Four Clusters in the Artificial Dataset - Ground Truth.

We ran the IMS with these 1000 InfoPacks for approximately 30 seconds on a standard PC. Figure 10 shows the results with the main clusters highlighted. The artificial noise in the data created more than four clusters, but the majority of the InfoPacks do fall closely together as expected.

6. Next Steps

Currently, we are working towards a large-scale evaluation of

the performance of the entire system with a group of actual users. In the one-week exercise, the users will be presented with two taskings each day (morning and afternoon) and they will execute their tasking against the open web using Google. The system will capture the contents of any web page that any of the users opens and create a document InfoPack for it. From the observed user actions, the UMS will create a dynamically changing user InfoPack. The IMS will recommend documents from the growing set of InfoPacks to other users that may execute the same tasking. Knowing which users are assigned the same taskings, we will analyze the user-to-user recommendations of the IMS to determine whether the emerging clustering of user InfoPacks is meaningful. Finally, the IMS will also re-rank on the fly Google search results for

specific user queries according to the user's current model, resolving ambiguities that are in the Google query (e.g., "Jaguar" could be a car or an animal) by knowing the user's interest.

7. Conclusion

Our Information Matching System self-organizes dynamically changing models of document contents and user interest, synergizing

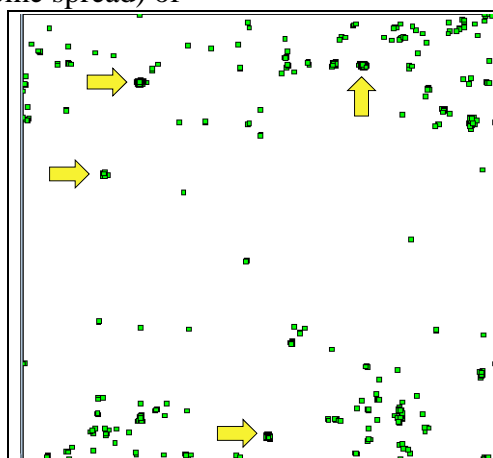


Figure 10. Clustering of 1000 Artificial InfoPacks.

multiple techniques for interpreting natural-language text, audio, or video contents that may have conflicting conclusions about the similarity of documents. By using sub-symbolic local and non-local interactions among simple agents in an abstract information space, the system is scalable, robust, and adaptive to change either in the models or the composition of the corpus. Alternative modeling and reasoning techniques may be brought on-line at any time, resulting in a graceful re-ordering of the user and document models to a new stable state. Recommendations of other users or relevant documents for a particular user may be extracted at any time. We are currently completing a comprehensive prototype of the entire analyst support system and will perform an evaluation with actual users soon.

8. Acknowledgements

This research was sponsored by the Air Force Research Laboratory, Air Force Materiel Command, USAF, under Contract number FA8750-06-C-0193. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

9. References

- [1] XMLBlaster.
<http://www.xmlblaster.org/>.
- [2] R. Alonso and H. Li. Model-Guided Information Discovery for Intelligence Analysis. In *Proceedings of Proceedings of CIKM '05*, Bremen, Germany, 2005.
- [3] S. Brueckner. *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*. Dr.rer.nat. Thesis at Humboldt University Berlin, Department of Computer Science, 2000.
<http://dochostrz.hu-berlin.de/dissertationen/brueckner-sven-2000-06-21/PDF/Brueckner.pdf>.
- [4] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Second ed. Chapman & Hall, 2000.
- [5] D. P. Dobkin, A. Hausner, E. R. Gansner, and S. C. North. Clustering force-directed graph layouts. In *Proceedings of 15th Annual Symposium on Computational Geometry*, pages 425-426, 1999.
- [6] R. Hilscher, S. Brueckner, T. C. Belding, and H. V. D. Parunak. Self-Organizing Information Matching in InformANTS. In *Proceedings of Self-Adaptive and Self-Organizing Systems (SASO07)*, Cambridge, MA, 2007.
www.newvectors.net/staff/parunakv/SASO07InformANTS.pdf.
- [7] L. Le Cam and G. L. Yang. *Asymptotics in Statistics: Some Basic Concepts*. Berlin, Springer, 2000.
- [8] M. Mamei, F. Zambonelli, and L. Leonardi. Cofields: a physically inspired approach to motion coordination. *IEEE Pervasive Computing* 3(2):52- 61, 2004.
- [9] S. Moujahed, O. Simonin, A. Koukam, and K. Ghédira. Self-Organizing Multiagent Approach to Optimization in Positioning Problems. In *Proceedings of ECAI 2006 - 17th European Conference on Artificial Intelligence*, pages 275-279, 2006.
- [10] H. V. D. Parunak, S. Brueckner, and J. J. Odell. Swarming Coordination of Multiple UAV's for Collaborative Sensing. In *Proceedings of Second AIAA "Unmanned Unlimited" Systems, Technologies, and Operations Conference*, San Diego, CA, AIAA, 2003.
<http://www.newvectors.net/staff/parunakv/AIAA03.pdf>.
- [11] P. Proulx, L. Chien, R. Harper, D. Schroh, T. Kapler, D. Jonker, and W. Wright. nSpace and GeoTime: a VAST 2006 Case Study *IEEE Computer Graphics and Applications*, 27(5):46-56, 2007.
- [12] E. W. Weisstein. Graph Embedding - From MathWorld--A Wolfram Web Resource.
<http://mathworld.wolfram.com/GraphEmbedding.html>