

# Autonomic Middleware and Resilient Information Systems

We describe work generated within a data and information fusion defence technology centre (DIF DTC) cluster project called 'Hyperion' [1]. The objective is to create an adaptive agent-based architecture capable of significantly enhancing the functionality and resilience of network-centric information fusion and Command and Control (C2) processes. The system supports the requirements of future Network Enabled Capability (NEC) operations, namely, intelligent and real-time reconfiguration of information and communication technology (ICT) services to meet bandwidth, hardware and human constraints.

This article focuses on the middleware component called 'Nexus', developed at BT, which provides resilient peer-to-peer messaging infrastructure and a knowledge support application 'Cyclone', which is used to categorise unstructured or semi-structured information from heterogeneous services, either hosted in the Nexus environment or from external sources.

By

Alex Healing, Hakan Duman and Robert Ghanea-Hercock

Centre for Information & Security Systems Research (CISSR), BT Innovate

Email: alex.healing@bt.com | hakan.duman@bt.com | robert.ghanea-hercock@bt.com

## Introduction

The problem facing both the security and defence communities is the volume, complexity and timeliness of information. In particular the ability to locate and access the right information at the right time is crucial to achieving real-time responsiveness and situational awareness. The Nexus system is a Peer-to-Peer (P2P) agent-based middleware that creates a fully distributed and highly resilient Service Oriented Architecture (SOA). The combination of a structured P2P overlay network and autonomous service discovery delivers a powerful capability to support network-centric command and control; reducing the need for staff dedicated to data transfer and training costs for end users, as well as being able to offer a new degree of accuracy and timeliness for information provision.

This paper outlines the overall architecture of the Nexus system and its application in a defence situational awareness scenario. A detailed review of the publish/subscribe methodology which allows users to define and subscribe to topics of interest is presented with particular focus on the proactive nature of Nexus for information and service delivery.

More specifically we explain how the system derives useful clusters of information on the network autonomously and learns from users' behaviour how best to categorise services. This work combines elements of human factors research for information visualisation with an underlying resilient information architecture, focused on making the infrastructure intelligent, such that the system acts as an effective broker between users and information on the network.

## Methodology

The Nexus middleware is based on three key paradigms: Service-Oriented Architecture (SOA) [2], Peer-to-Peer Computing, and Autonomous Agents; all of which have been identified as key components of future NEC network architectures [3, 4, 5].

The SOA approach acts to connect distributed sources and users of information, dealing with the integration of legacy systems and reducing complexity by wrapping and abstracting resources as uniform services. Most implementations of SOA, however, are to a large extent pre-engineered and centralised in their infrastructural implementation. Core services for enabling

an SOA, such as discovery, are typically controlled by a single centralised entity, which not only results in a rigid solution but also a highly vulnerable one in domains where attack is likely.

In the design of Nexus, we have sought to decentralise the SOA enabling infrastructure, allowing service discovery, for example, to be performed without reliance on a central registry so as to remove any potential single point of failure. The research challenge when decentralising such a large-scale system is effective management in a distributed context. We employ Autonomic Computing [6] to help manage the complexity and design a system of collaborative software agents embedded within the Nexus middleware which perform service discovery and semantic selection tasks [7]. Using the peer-to-peer overlay as a substrate, these agents learn about the changing supply of information services on the network as well as users' requirements, and adapt to suit.

## Architecture

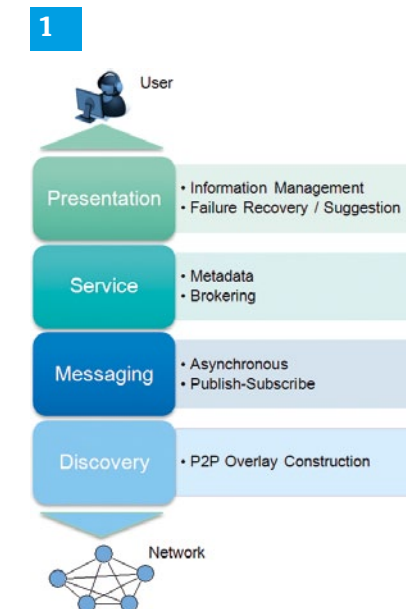
The Nexus middleware is intended for deployment on any number of distributed platforms and, as it is implemented in the Sun Java programming language, it is not dependent on the operating system environment. An instance (peer) of the middleware is deployed on each machine in the network and, assuming there is physical connectivity, the peers automatically configure to form an overlay network. Figure 1 shows the layered architecture of a single Nexus peer.

Phase I of the Nexus project was focused primarily on the lower two levels in Figure 1. It was established at this early stage that the challenge to be tackled was how to make a completely decentralised model of discovery effective in the large-scale distributed and dynamic environment that NEC mandates. The aim has always been to create a resilient yet highly useful medium for a user of the system, in order to drive information fusion.

Phase II has seen the focus of research move to the top two levels where there are far more possibilities to give the system an understanding of semantics and thus engineer autonomic behaviour [6, 7]. A reliable, standards-based approach to offering a discovery and messaging substrate has been developed, based on the work conducted in Phase I, which is used by the higher-level autonomic components. An example is the use of a collaborative multi-agent system to collectively model which network services are best suited for particular

users [7]. In this case, the agents form part of the Service layer of Nexus but rely on the Messaging layer to disseminate quality-of-service information between one another, and that in turn relies on peers having discovered each other in the first place. The autonomic components form the majority of the novel research undertaken and several patents have been filed in the areas of decentralised security, multi-agent systems, and the semantic web.

It should be noted that although the higher levels are dependent on the functionality of the lower levels, they are at the same time agnostic to how exactly the lower levels have been implemented. For example, in the current implementation of the Nexus middleware we adopt a fully decentralised approach at the discovery layer, but this could be replaced with a centralised approach without affecting the layers above. The rest of this section describes each of the four layers in more detail.



The Nexus Peer Architecture.

## Discovery layer

The discovery layer's prime function is peer discovery, such that a platform on the network can determine the presence of other platforms. Each Nexus peer broadcasts its presence on a shared channel which allows peers to collate the endpoints to all other peers. This so-called *worldview* of the peers forms the basis of the peer-to-peer overlay network and is maintained such that new peers are discovered automatically and those leaving the network are removed from the worldview. The presence

information is minimal in size enabling the approach to scale well and, unlike many alternative peer-to-peer discovery techniques, allows for a complete view of the network.

#### Messaging layer

As the underlying peers may be highly dynamic in their availabilities, when designing Nexus, rather than focusing on direct messaging between peers, we allow abstraction to *topics* of message by implementing a publish-subscribe methodology. We use a decentralised implementation of the Java Message Service (JMS) specification to manage the message routing. Typically, message-oriented middleware and indeed most enterprise uses of JMS rely on a centralised message broker. In the version we employ, each peer in the overlay network has the ability to act as a broker. This ensures that there is no single point of routing failure, if there is a failure it is localised and potential bottlenecks are avoided.

Each message in Nexus is published on either one or many topics. The topics form a hierarchy where sub-topics of existing topics can be defined. When a user wishes to receive messages they subscribe to either one or more topics and will then asynchronously receive the messages published to the topics until they choose to unsubscribe. The challenge is how the topics should best be defined, such that they are tailored for the collective of users and the NEC requirements, i.e. allowing access to the right information 'at the right place, at the right time – and not too much'. It is this challenge which has been the focus of research and is elaborated on in the rest of the paper.

#### Service layer

The service layer builds upon the messaging layer by using it to disseminate information about the platforms in the network, the services that they host, the structure of information in the network, as well as the service content itself. This layer represents the core of the SOA approach and allows for all Nexus peers to be able to *host* services. Services in the context of Nexus are typically information sources where the default mode of interaction is *pull*, however, services with a more complex interaction could also be incorporated, for example computational services that offer some processing capability. Fundamentally, though, any service has some description which needs to be disseminated to all interested parties on the network, such that they can discover its presence and interrogate it for further information.

The foundation of the service layer is the format of the messages sent between Nexus peers. There are three different Nexus message types: Host Advertisements, Service Advertisements, and Topic Advertisements. These advertisement type messages are typically used for notifying subscribing peers of the presence and description of other Nexus Peer's logical hosts (e.g. where the platform running the Nexus peer is located), their hosted services (information sources being published from the platform) and the current map of topics (subscription channels which are currently being published to).

Achieving effective service discovery without a centralised registry of some form is a non-trivial problem. Much the same way as peer discovery is achieved at the bottom layer of Nexus, the logical descriptions of platforms and their services are kept track of by each peer having their own 'view' of the 'world'. This equates to each peer having their own registry which they populate from the advertisement messages that they receive. Keeping this view of the other entities in the network up-to-date requires a significant amount of traffic and thus the point of partitioning the message space into topics is reiterated. Such partitioning of the message space depends on knowledge about the information providers and the actual information content, such that similar relevant information sources can be clustered and published on specific shared topics.

When deciding on a suitable framework for the metadata of the services within Nexus, which is used to describe both the information provider and to some extent the information content, the following concerns were identified as being important:

- Expressive: there should be sufficient coverage of the functional attributes of the information provider and the information itself.
- Lightweight: due to the challenges imposed by adopting a purely decentralised environment and a potentially bandwidth-restricted NEC environment, the metadata must be as lightweight as possible, such that the resultant advertisement message sizes remain small.
- Extensible: due to the dynamic nature of a service's description and the frequency of update for different parts (e.g. the location of a mobile service altering more frequently than the service's overall capability), it is occasionally important to send updates for only those parts of the description that have changed.

- Human *and* Machine readable: the initial process of mark-up of a service need be human-friendly enough that it doesn't require a huge amount of effort. Complementing the notion of empowering the fringes of the network that the P2P approach introduces, it may be realistic to allow the service consumer the authority to alter the mark-up of services. Making sure that the metadata is machine-readable is clearly necessary for any autonomic system to then act upon it, e.g. fusing and orchestrating services on behalf of a user.

Given these constraints a description schema based on eXtensible Markup Language (XML) was developed. This is a relatively primitive technology compared to more recent attempts at metadata languages such as the Resource Description Framework (RDF), or Web Ontology Language (OWL) [7], however these more advanced formats introduce additional complexity due to their verbosity and machine-oriented focus. Such standards employed in typical Semantic Web [8] efforts, in particular ontological languages such as OWL, also mandate a high level of pre-engineering. Instead, by remaining simple, the format chosen for Nexus is lightweight whilst flexible and is capable of scaling to both the network complexity and time constraints of the intended deployment environment.

We have sought to define a description schema which, although based on XML, does not solely consist of structured and explicit terms – for example RDF statements consist of well-defined triples. The motivation being that the construction of well-defined and structured concepts is a costly process and difficult to adapt. Naturally, without having some elements of the metadata structured in advance makes processing upon the elements unrealistic; and indeed attributes such as location, for example, have been made explicit in our schema to achieve inference on certain aspects. The focus of our research, however, has been to investigate the degree to which an unstructured portion of the metadata may be useful with regard to both human and autonomic interaction with the description. Of particular interest is whether it is possible to effectively use both structured and unstructured metadata in conjunction, such that an explicit goal-based capability as well as flexibility is achieved from each of the approaches respectively.

Our approach is aligned with the trends on the Web over the past few years whereby the vision of the Semantic Web is not yet a reality and despite numerous standards being drafted,

none have experienced widespread adoption. The growing trend of lightweight 'tagging' for marking up resources has instead become popular and forms the basis of searching and indexing within most large modern websites regarded as constituting what is now widely known as Web 2.0. Nexus thus allows service providers, or indeed service consumers, to manipulate the mark-up of the services by changing the set of tags describing them. The tags themselves act as a rough categorisation and the approach of collaborative tagging is often called *folksonomy* – a combination of folk and taxonomy.

A 'Categories' field in the Nexus service advertisement message is used as a more concrete notion of categorisation than tags and may only be manipulated by an authorised administrator of the information network, whereas the policy for changing tags may be laxer. Typically in information systems, the structure of the categories is pre-defined, usually by means of an ontology, however, Nexus assumes no such prior knowledge or agreement. Instead, it uses a mixed-initiative process to categorise services based on their informal sets of tags supplied by the collective of users.

We are not proposing that such a lightweight methodology is the answer to the problem of resource description in its entirety; however it may be short-sighted to assume that an approach of pre-engineering an ontology of resources, for example, is a complete solution for all cases [9]. Making such a description framework based on folksonomies scale to large, complex systems by 'distilling' them, from an informal set of tags to a concrete taxonomy, is a challenge that we have addressed with our research into the *Cyclone* application later in this article.

#### Presentation layer

The purpose of the presentation layer of Nexus is to interface with the user and as such be information-centric, abstracting away from much of the lower functionality. It is in this layer that the autonomic capability of Nexus is being primarily developed, however, the user is retained in the loop rather than letting the system-based automated reasoning function purely alone. Instead, we have designed a system that can accept feedback from users and learn; hence adapting so as to improve its automated behaviour.

The typical approach to fulfilling an information requirement of a user is to formalise both the goal and match that with a

formalised and elaborate service descriptor. For the reasons discussed in previous sections, Nexus has neither a very elaborate nor formalised service descriptor. Nevertheless, if the system is able to reason about how similar other services are to each other then it could, for example, offer similar information to that which a user is already browsing. Failed or degraded services can be substituted with relevant alternatives using the same logic.

Rather than assuming that similarities as a result of analysing the service metadata are absolute, we have developed a tool called *Cyclone*, which incorporates automated clustering algorithms. It involves the user and learns from their actions so as to autonomously categorise services. Once categorised, services are advertised in publish-subscribe topics matching the name of each of their categories, allowing users to subscribe to all sources of a particular category of information. The Cyclone system is described in detail in the following section.

### 'Cyclone' categorisation

The Cyclone application, developed alongside Nexus, is used to formally categorise the services that have metadata based on sets of unstructured tags. This allows information providers to supply merely a lightweight, tag-based description of their services, and an authorised system administrator, for example, to organise the information network, updating those informal service descriptions and form an emergent taxonomy. This is particularly important when the service descriptions are highly dynamic and thus mandate a fluid taxonomy, as well as those which have been derived automatically by a keyword extraction algorithm.

The on-the-fly nature of categorisation by the Cyclone system makes it possible to incorporate and structure information created internal to operations, e.g. intelligence created in theatre in a military scenario; as well as external information, e.g. from open sources on the Web. For both cases only a limited amount of metadata is necessary, but rather than relying on this metadata alone to advertise and discover the information, Cyclone allows this to be refined into more formal and usable categories. Figure 2 illustrates this at a conceptual level; where a 'tag-cloud' on the left-hand side, the most common visualisation of tagged information systems on the

Web, which can be useful for some systems to gain an overview but typically does not scale sufficiently, is contrasted with a formal tree of categories where each one contains a set of the information sources available and relationships between the categories can be defined.

A screenshot of the Cyclone application is shown in Figure 3 and demonstrates the juxtaposition of information services from the Nexus middleware system, as well as incorporating feeds from the Internet. Information sources are displayed in the centre panel and visual clustering illustrates how similar they are derived to be. Before any user interaction, this similarity analysis is based solely on the metadata supplied, but as a user creates categories and assigns the data to them, the system adapts its clustering to reflect the changes.

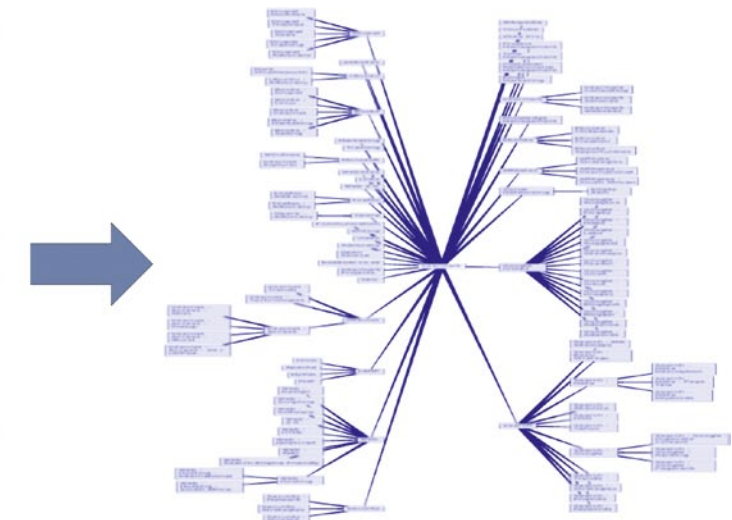
We have sought to automate the process of categorisation based on service metadata by employing well-known unsupervised clustering techniques but in order to keep the user in the loop as much as possible, an important requirement for our system at design-time was that the output from this clustering be presented at all times to the user. This is achieved by translating the similarity values, derived from cosine similarity analysis, into forces between each of the data nodes and representing them as a force-directed graph. The resultant graph with edges removed is what makes up the majority of the centre panel of the application.

In order to enhance the user-centric nature of the system, the unsupervised clustering algorithm is augmented with supervised learning based on a neural network, which adapts the forces to improve the clustering accuracy as users interact with the system. The result is that over time the system adapts such that its automated behaviour is increasingly accurate at categorising data, becoming personalised for a particular user or group of users. Given this knowledge, Cyclone can then autonomously assign new information to categories.

The Cyclone system was demonstrated at the Coalition Warrior Interoperability Demonstrator (CWID) UK 2008 where information feeds from other demonstrator partners were aggregated and categorised. For further technical information on the Cyclone system we refer the reader to the following papers: [10, 11].

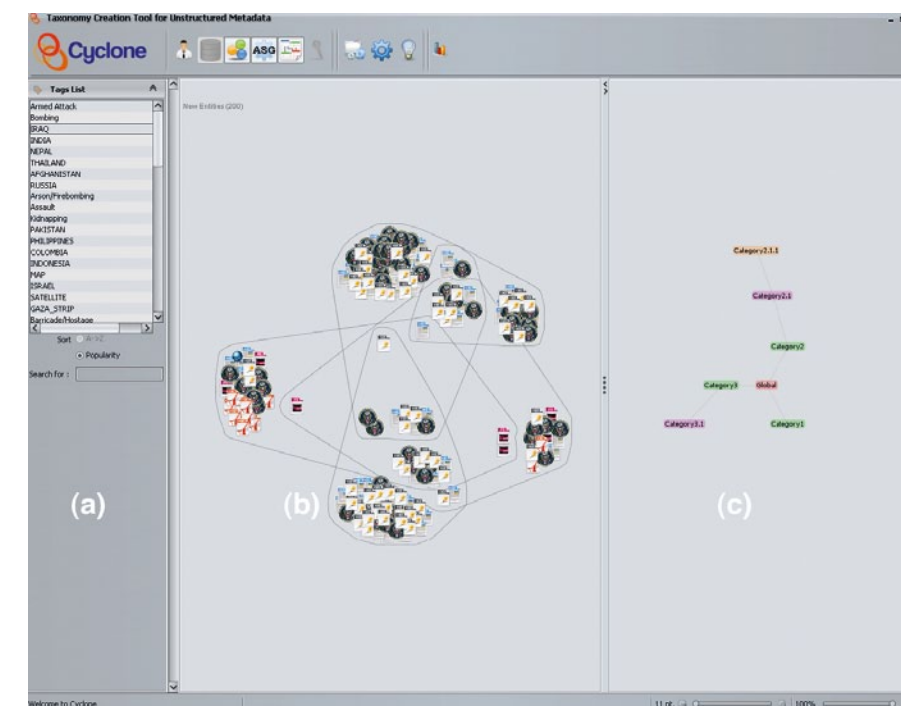
2

accountability address address adjustments administration algeria america american announced appropriations apr army assistance aug authority awarded briefing bush care central challenges coalition command committee conference congress construction continuing contracts council country dec defense department deploy destruction director discusses documents dod economics education elections emergency estimate fact feb forces freedom general goals government health house issued jan john joint jul jun kb law major management mar mass mb measuring meets military minister mission national nov occupation oct office oil operation pdf people policy political power president press prime program progress provide provisional public pursuant quarterly rebuilding reconstruction related released relief remarks report resolution rights saddam secretary section security senator senate sep services sigir special specific stability staff statement states strategy submitted support terror testimony today transcript transition through un united update version war washington weapons weekly world

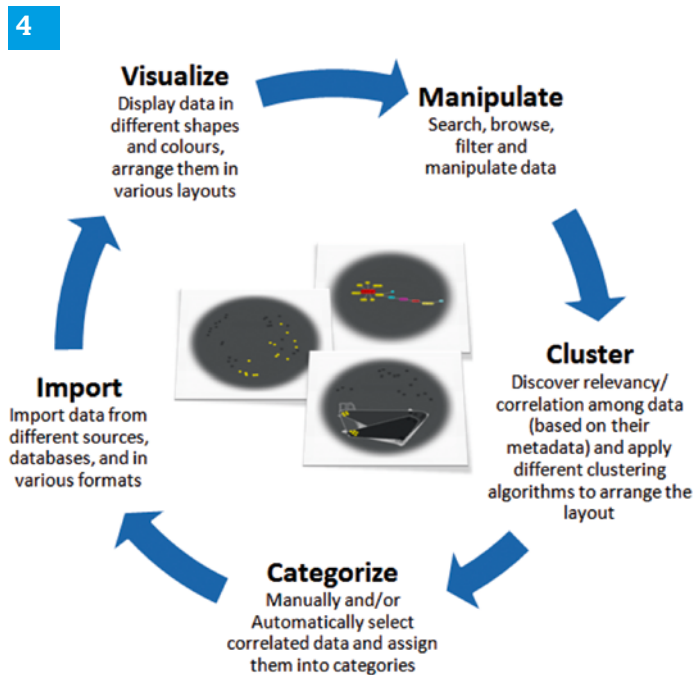


The overall concept of Cyclone is to allow for an improvement to the descriptive vocabulary of folksonomies in Web 2.0 systems and improve upon the commonly used Tag Cloud interaction (left-hand side) by creating a more formal Taxonomy (right-hand side).

3



Cyclone Graphical User Interface: (a) List of tags for all services discovered by Nexus, (b) Clustering Panel showing similarity between services as fuzzy clusters (grouped icons), (c) Taxonomy Panel showing the hierarchical structure of categories which may be either predefined or constructed by users as a result of the interaction with the Clustering Panel.



The iterative process of interaction with the Cyclone system.

### Conclusion

The aim of Nexus and Cyclone is to make the supporting technology 'invisible' to the end user. The value is to reduce the cognitive workload of the user and to make dynamic service provision as seamless as possible, without detracting from the users' local knowledge or forcing constrained solutions on them. The philosophy is 'just enough AI' to help bridge disparate systems and services together, but not so much that the solution becomes part of the problem; as is so often the case in large-scale ICT systems.

The resulting value to the NEC process is reduced manpower to support ICT systems, and reduced training requirements to operate large-scale C2 NEC systems. The value to the field commander is a true 'plug-and-go' ICT capability.

### References

1. R. Ghanea-Hercock, E. Gelenbe, N. R. Jennings, O. Smith, D. N. Allsopp, A. Healing, H. Duman, S. Sparks, N. C. Karunatillake, and P. Vytelingum, 'Hyperion Next-Generation Battlespace Information Services', *The Computer Journal*, 2007.
2. M. N. Huhns and M. P. Singh, 'Service-Oriented Computing: Key Concepts and Principles', *IEEE Internet Computing*, vol. 9, no. 1, pp.75–81, 2005.
3. A. Alston, 'Network Enabled Capability – The Concept', *Journal of Defence Science* 8(3), 108–116, 2003.
4. S. Manoski, 'Eliminating the Middleman, Peer to Peer technology for Command and Control', [http://www.mitre.org/news/the\\_edge/summer\\_02/manoski.html](http://www.mitre.org/news/the_edge/summer_02/manoski.html)
5. R. Prouty, K. Kalbus, and L. Lee, 'Peer-to-Peer Discovery: A key to enabling robust interoperable C2 architectures, C4ISR/C2 Architectures', *The Tenth International Command and Control Research and Technology Symposium Virginia*, 2005.
6. J. O. Kephart and D. M. Chess, 'The Vision of Autonomic Computing', *IEEE Computer*, 36(1):41–50, 2003.
7. S. A. McIlraith, T. C. Son, and H. Zeng, 'Semantic Web Services', *Intelligent Systems*, Special Issue on the Semantic Web. 16(2):46–53, March/April, 2001.
8. T. Berners-Lee, J. Hendler, and O. Lassila, 'The Semantic Web', *Scientific American*, May 2001, pp.28–37.
9. C. Shirky, 'Ontology is Overrated: Categories, Links and Tags', 2005, [http://www.shirky.com/writings/ontology\\_overrated.html](http://www.shirky.com/writings/ontology_overrated.html)
10. H. Duman, A. Healing, and R. Ghanea-Hercock, 'An Intelligent Agent Approach for Visual Information Structure Generation', 2009 *IEEE Symposium on Intelligent Agents*, April, 2009.
11. H. Duman, A. Healing, and R. Ghanea-Hercock, 'Adaptive Visual Clustering for Mixed-Initiative Information Structuring', to appear *HCI International 2009 Conference*, July, 2009.