24-Hour Knowledge Factory: Using Internet Technology to Leverage Spatial and Temporal Separations

Amar Gupta

Thomas R. Brown Professor of Management and Technology, University of Arizona

Satwik Seshasai

Software Development Manager, IBM, and Doctoral Student, MIT

Abstract

A “24-Hour Knowledge Factory” involves a team distributed across three or more collaborating centers connected to each other by internet technology or by dedicated networks, with work on specific endeavors being performed on a round-the-clock basis. A white-collar professional could work in the US on a standard 9 am to 5 pm basis; at the end of his or her workday, the activity could be transferred to a colleague in Australia who works during daytime in that country; and, at the end of the latter’s workday, the work could be transferred to a third colleague in Poland, who in turn could pass the baton 8 hours later to the first worker in the US. In this scenario, each member of the team would work during the normal workday hours that pertain to his or her time zone. The effective use of sequential workers in a 24-Hour Knowledge Factory requires that professional tasks be broken down to the level that individuals can work on them with minimal interaction with their peers. In addition, one requires new technological approaches that would reduce the effort involved in transitioning from one employee to the next. The latter aspect is facilitated by the concept of composite personae. A pair of technology prototypes was developed; these prototypes leverage internet-based
capabilities to redefine the manner of sharing of knowledge. One prototype uses a web-based interactive system coupled with a unique data model to optimize collection and storage of design rationale and history from both stakeholders and workers. The other prototype presents an interaction model where multiple individuals can act as one “composite persona” when interacting with these systems on the internet, thereby allowing tasks and knowledge to be shared across the internet in a seamless manner, without the need for complex authentication and security models. The combination of these prototypes provides the foundation for an integrated internet-based system for implementing the 24-hour knowledge factory model. Further, a case study was conducted at IBM to observe the harbinger of a 24-hour knowledge factory in action and to determine the role that internet technology played in accomplishing the overall endeavor.

1. Introduction

“The Sun never sets on the British Empire,” was a notion emphasized during the eighteenth and nineteenth centuries to highlight that the British Empire was far-flung, and that the sun was always visible from some part of this vast empire. While the British Empire has gradually disintegrated, we can now coin an equivalent notion: “The Sun never sets on the 24-hour Knowledge Factory!”

The term “24-Hour Knowledge Factory” connotes a global delivery model in which members of a global teamwork on a project around the clock; each member of the team works the normal workday hours that pertain to his or her time zone. At the end of such a workday, a fellow team member located in a different time zone continues the same task.
In a sense, the notion of the 24-hour Knowledge Factory originated in the concept of continuous assembly line, an outgrowth of the industrial revolution. Since the installed equipment was scarce and costly, the employees were scheduled to work in shifts, in order to use the manufacturing facilities on a round-the-clock basis. With the advent of electronic computers and diminishing costs for telecommunications, one developed the notion of 24-hour Call Centers. Depending on the time of the call, it is automatically directed to a call center that is active at that time. With three or more call centers located in appropriate time zones, one can ensure that all employees of these geographically distributed call centers are working during daytime in their respective countries. The notion of multiple support centers was subsequently adapted for supporting global communications networks, and can now be applied for professional tasks that are higher up in the value chain.

Note that the concept of the globally distributed call centers became feasible by the advent of inexpensive Voice over Internet technology. In fact, Internet technology is fundamental to allowing white-collar work to flow from location to location and to allow individuals in the system to work during the hours they are most effective (usually 9 am to 5 pm). The use of internet technology is also the key enabler to having workers perform certain minimal tasks and communication during the night. By having three sets of individuals use internet technology to exchange components of tasks over a 24-hour period, the time needed to produce knowledge-based assets can be substantially reduced.
Further, the concept of 24-Hour Knowledge Factory applies to many types of knowledge-based industries, not just to the field of information technology.

With respect to the realm of internet technology, two aspects are relevant. One is the pivotal role played by internet technologies to extend work to a 24-hour period. The second is the potential to use the 24-hour Knowledge Factory concept to facilitate the creation of next-generation internet technologies. For example, by involving specialized microchip design engineers located at multiple places around the world, a semiconductor chip design firm can create virtual “24-hour knowledge factories”. This provides the firm with access to high-talent designers who would otherwise have to move to a different country, or work at odd hours of the night; some persons call the latter type of shift as the “graveyard shift”. Overall, new chips can be designed and manufactured in much shorter periods of time; these chips, in turn, can be used to create succeeding versions of the regional and national infrastructure for internet. Accordingly, internet technology is both the driver and a beneficiary with respect to the 24-Hour Knowledge Factory framework.

Whereas a manufactured item was the end-product in the case of the “24-hour factory” that emerged as a consequence of the “industrial revolution”, knowledge-based services and knowledge-based products are the end-deliverables in the case of the current “information revolution”; hence the term “24-Hour Knowledge Factory”. The use of the term “factory” emphasizes: (i) the use of multiple sets of workers; (ii) the availability of a set of base technologies to operate on the knowledge and to transfer the knowledge from one worker to another; (iii) the shift-type nature of operations; and (iv) the underlying
idea of decomposing a big task into a series of components that can be tackled on a sequential basis.

Many industries are characterized by a development cycle that relies heavily on sequential performance of specific functions, such as development, testing, and verification. In a traditional software development environment where all parties are located in the same geographic area, a code developer typically waits until a fully functional portion of the product is available before passing it on to an engineer to test it. However, with the potential for receiving testing feedback overnight, the developer now has the unprecedented opportunity to build and test portions of the product on an incremental basis. Until the eighties, even the early nineties, people across the world perceived the time difference to be a major impediment. Now that has switched around - for many projects, the time difference can be a significant advantage.

The 24-hour Knowledge Factory will involve “offshoring” of part of the endeavor. Today, offshoring is done primarily to reduce costs. We believe that over time, the growth in offshoring will be fueled by the potential to achieve drastic reductions in turnaround times for major endeavors. The ability to transcend geographic and temporal boundaries offers the potential to change the face of many industries. This innovation will dramatically impact the manner in which companies build, test, sell, and support their products and services. In this paper, we look at the key attributes of the 24-Hour Knowledge Factory, and the pivotal role that Internet Technology plays with respect to this emerging paradigm.
2. Characteristics of 24-Hour Knowledge Factory

Figure 1 shows a continuing experiment involving faculty and students in three countries: the Wroclaw University of Technology in Poland (WUT), the University of Arizona, Tucson, USA (UA) and the University of Technology, Sydney, Australia (UTS). In such an operating environment, the distributed personnel can work on an incremental basis on the same task. Alternatively, each geographic location can be responsible for a separate task, and the overall efficiency of the project would be still improved in comparison to a single site basis; this is because in a tri-site scenario, each location perceives that progress is made “overnight” when workers at that location are asleep.

![Figure 1: A 24-hour tri-foci scenario.](image)

Note that it is not sacrosanct that each collaborating center performs work on a problem for exactly eight hours at a stretch; the involvement could be for shorter or longer durations of time. Further, an individual task does not need to be completed in less than one shift for it to be useful to the overall endeavor. But, one does need mechanisms for transferring work-in-progress to the next center with speed, efficiency, and low overhead.
Further, one does need methods for decomposing the primary endeavor into pieces that can be handled in a modular manner.

The success of the 24-Hour Knowledge Factory model relies heavily on a number of concepts that are central to the field of internet technology. Existing models for determining location of distributed factories, developed primarily for the manufacturing industry, emphasize that the availability of a distributed and reliable network across multiple plants is a critical success factor. When these models are considered in conjunction with frameworks for organizing decision making structures in a global organization, one finds a high need for having trusted, secure and private systems to manage the knowledge being shared within and across organizations. After the geographical decisions are made, the next important step is to incorporate decision making and knowledge management technologies for geographically and temporally distributed locations, so that the “plants” can operate in a seamless and productive supply chain. However, existing literature has generally neglected to treat the internet-based delivery system as a knowledge factory, where knowledge is the key component that is produced and leveraged across the internet.

We mentioned briefly about call centers earlier. Even though the work in call centers is “white-collar” in nature, it bears lot of similarity to blue collar work. In other words, the work is inherently very structured. At the other end, the work performed by President George W Bush and other heads of nations is frequently crisis-driven and is inherently ill-structured. As shown in Figure 2, the concept of 24-Hour Knowledge Factory is aimed
primarily at the intermediate territory of semi-structured work. Instances of semi-structured work arise in virtually all types of professional endeavors that involve mental work. This includes professional work from medical, logistics, product design, finance, accounting, and legal arenas. Further, the work does not have to be office-based.

Figures 2: 24-Hour Knowledge Factory Concept Applies Primarily to Semi-Structured Professional Work

In general, the 24-Hour Knowledge Factory paradigm is appropriate for situations where the professional endeavor can be broken down into components, the underlying knowledge can be digitized, different individuals can potentially work on such components with minimal support from their peers, and the work-in-progress can be transferred at minimal cost from one collaborating center to another.

The process of decomposition of work needs to be clarified here. In the production of intellectual assets, outsourcing has traditionally been accomplished by decomposing a large task into smaller subtasks. This "horizontal" decomposition creates subtasks that are
highly cohesive but minimally coupled to sibling tasks; as such, a rather high degree of autonomy can be applied to solving each task. In contrast, the 24 hour knowledge factory may frequently involve vertical decomposition of tasks with multiple persons operating on the same task in a sequential manner. Now, collaboration shifts from the "minimal coupling" to the "high cohesion" side. This shift necessitates more knowledge transfer and more intimate communication between the collaborating parties than in the case of the horizontal decomposition.

3. Relevant Internet and Allied Technologies

The 24-Hour Knowledge Factory is a distributed enterprise driven by a number of evolving technologies, as exemplified below.

**Collaborative Technologies:** With the widespread adoption of the Internet, collaboration technologies have evolved too. These collaboration technologies provide efficient mechanisms for the transmission of textual, numeric, pictorial, and other types of information, all at very low costs. Major software vendors in the enterprise collaboration space have built on the notion that most workers are constantly connected via the Internet and have embedded collaborative technology directly into office productivity software. For example, both Microsoft’s SharePoint technology and IBM’s Quickr technology allow knowledge workers to share information directly from Microsoft Word, Excel, and PowerPoint without having to first save the file locally and then to transmit it via electronic mail or web-based upload. These advances are indicative of the need to have
internet supported collaboration directly integrated within native applications if the 24-hour knowledge factory is to operate unobtrusively to the worker.

**Wiki:** While electronic mail is used primarily for relatively short messages, new collaborative technologies are emerging for sharing larger information and knowledge objects. The Wiki Web (or simply, wiki) is an example of website software that allows visitors to easily add, remove, or otherwise edit and change available content. Wikis require little in the way of computing resources making them ideal for small workgroups to share information on unresolved problems and documentation on pertinent issues. In addition, wikis are equipped with embedded version control and search engines, and can create hyperlinks and content indices. While most wikis are small, Wikipedia, the open Internet encyclopedia has more than 1.5 M articles, demonstrating the scalability of the wiki concept. In globally distributed companies such as IBM, wikis serve as a platform for informal knowledge sharing among the collaborating teams [Hill, C. et. al. 2006]. The key technical innovation behind the wiki is the ability for any individual viewing a document to edit it in-place, and have the edits be reflected immediately in the online version. Previous versions are saved so that malicious or incorrect information can be reverted easily. This technology is indicative of the need for open internet technology that can provide good support, but at negligible overhead, to the 24-hour knowledge factory worker.

**WEB-DAV:** Web browsers, used to access the World Wide Web, can also be used to access information in private networks and content in file systems. Distributed authoring
and versioning systems, like WEB-DAV [Whitehead, 1997], enable distributed web-authoring. For overwrite protection, WEB-DAV provides an exclusive-write and a shared-write lock. The extended HTTP1.1 will provide a standards-based infrastructure for asynchronous collaborative authoring on the web [Fielding R. et al., 1997].

**Version Control:** With multiple centers, sometimes located in nations with unreliable electronic communications infrastructure, companies opt to replicate information repositories at each site. The need to keep each repository synchronized with each other requires the use of version control. The concept of version control is also utilized to keep track of succeeding versions of the information systems. In the old days, the waterfall model was used as the basis for systems development; one version of the software was completed before work commenced on the next version. Today, the succeeding versions are dovetailed. Companies like Intel and Motorola design new semiconductor chips in parallel, though slightly shifted in time with other concurrent activities [Tan, et al. 1996]. Changes made in program code, documents, diagrams, and other files at one site within the 24-Hour Knowledge Factory will need to be replicated at other sites. Examples of version control systems that could be used for this purpose are RSYNC, CVS, and Subversion [Cederqvist, et al., 2005].

RSYNC is an algorithm, periodically invoked, to keep two or more file systems synchronized [Tridgell, 2000]. RSYNC accomplishes synchronization using block-level checksumming to create information deltas between two files. Over the network, only the incremental changes are communicated, permitting the use of relatively low
bandwidth lines found in many developing environments. The Concurrent Versioning System (CVS) allows two or more developers to work concurrently on the same files [Grune, 1986]. Ordinarily, concurrent editing leads to one developer overwriting another's previously committed changes, an obvious impediment to large-scale development: CVS overcomes this problem. CVS is being gradually replaced by Subversion that permits transmission of incremental changes in both directions.

**Extreme Programming (XP):** Extreme Programming advocates collective ownership of code, allowing anyone in the project to alter any artifact [Beck, 2000]. This programming concept becomes especially relevant in the case of collaborating work centers where multiple individuals will modify and expand the code at different times of the day. Unlike the conventional paradigm in which each individual focuses on the assigned tasks and interacts with peers only on an exception basis, a programmer working in XP environment knows that others will be manipulating the same piece of code too. XP places heavy emphasis on automated testing. These results can be relayed, via Internet, to provide knowledge to all co-workers about the current and anticipated state of the task. Xiaohu et al. [2004] describes the use of XP in a case involving globally dispersed teams.

**4. Security in Internet Environment**

Individual sites within a 24-Hour Knowledge Factory environment will utilize the Internet to communicate all types of information including sensitive personal information (such as medical and financial information) and valuable intellectual property (such as for
new product design or drug design). A combination of evolving encryption, digital signatures, and authentication techniques is necessary to secure data in the factory. Encryption ensures data are protected; signatures ensure that the claimed sender of the data is the actual sender, and authentication ensures that the recipient is the intended recipient.

**Encryption:** Encryption is the process of encoding data, prior to transmission, in order to make them more secure. The National Institute of Standards and Technology (NIST) of the US Government has defined standards that have become de facto standards across the Internet. Replacing the aging Data Encryption Standard (DES), the new relevant standard is the Advanced Encryption Standard (AES). AES incorporates a symmetric (shared) key cryptosystem, with key entropies of 128, 192, or 256 bits [AES, 2001].

Public key cryptography [Diffie and Hellman, 1976] differs from symmetric key cryptosystems in that half of the encryption key is publicly known. To encrypt a message from one party to another, the sender encrypts the message using the publicly known key for the particular recipient. After the message is encrypted, the receiver can use the secret half of the key to decrypt the message. This kind of cryptosystem is used to exchange a randomly generated session key between anonymous parties. This session key is subsequently used to encrypt further communication using a symmetric key cipher, such as the aforementioned AES. This is essentially the process used by the Hyper Text Transfer Protocol Secure (HTTPS) to securely encrypt web transactions between a client's browser and the remote server.
IPsec attempts to implement security at the IP level, and can protect one or more "paths" between a pair of hosts, a pair of security gateways, or a security gateway and a host. The term "security gateway" refers to an intermediate system that implements IPsec protocols. Virtual Private Networking (VPN) is a similar technology that encrypts a channel between two hosts using public networks. VPN is a private data network that makes use of the public telecommunication infrastructure, and maintains privacy through the use of a tunneling protocol and security procedures. There are three important VPN technologies: trusted VPN, secure VPN, and hybrid VPN.

**Digital Signatures:** Digital signatures are used to verify the authenticity of data especially in situations where documents and data structures are transmitted over insecure networks. Digital signatures use one-way hash functions to cryptographically compress a data object into a small signature. This signature can then be encrypted. To verify the authenticity of a document, the verifier decrypts the signature and compares the decrypted hash to a hash of the document in question. If the two hashes are equivalent, then the document is verified to be authentic. The NIST standard, SHA (Secure Hash Algorithm), generates a 160-bit hash [SHS, 2002].

**Authentication:** Authentication is the process of guaranteeing the identity of a remote entity. The Lightweight Directory Access Protocol (LDAP) can be used for accessing distributed directory services that act in accordance with X.500 data and service models. An LDAP directory reflects various political, geographic, and organizational boundaries.
LDAP supports referrals and can be used to construct federated authentication systems such as those envisioned for the 24-Hour Knowledge Factory.

5. Internet Technology Foundations of the 24-Hour Knowledge Factory

In building this 24-Hour Knowledge Factory model, a number of known ideas and concepts can serve as valuable building blocks, as summarized in the following table:

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internet Technology Global Development</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carmel and Agarwal [2002]</td>
<td>The Maturation of Offshore Sourcing of Information Technology Work</td>
<td>Separability of software production has reduced transaction costs for coordinating IT work.</td>
</tr>
<tr>
<td>Aron and Singh [2004]</td>
<td>IT Enabled Strategic Outsourcing</td>
<td>IT work is on a graduated knowledge continuum, and information workers are vital at each stage.</td>
</tr>
<tr>
<td>Carmel and Agarwal [2001]</td>
<td>Tactical Approaches for Alleviating Distance in Global Software Development</td>
<td>Software development projects in the future will involve separated centers of engineers spread around the world with individuals dispersed remotely.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Using the Internet for Knowledge Management</strong></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lei and Slocum [1992]</td>
<td>Global strategy, competence building and strategic alliances</td>
<td>Distributed databases must be used to share core competencies’ across sites.</td>
</tr>
<tr>
<td>Powell [1998]</td>
<td>Learning from collaboration: Knowledge and Networks in Biotechnology and Pharmaceutical Industries</td>
<td>Knowledge creation is a core competency – it requires productive collaboration over the internet.</td>
</tr>
<tr>
<td>Davenport [1999]</td>
<td>Managing customer support knowledge</td>
<td>Internet technology can not replace human input into the system with automation.</td>
</tr>
<tr>
<td>Quinn [1999]</td>
<td>Strategic outsourcing - leveraging knowledge capabilities</td>
<td>Effective internet technologies will eliminate duplication of efforts.</td>
</tr>
<tr>
<td>Gupta [2001]</td>
<td>A Four-Faceted Knowledge Based Approach to Surmounting National and</td>
<td>Technology for sharing knowledge across borders requires focus on knowledge</td>
</tr>
</tbody>
</table>
### Other Borders

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>Key Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacCormack et al. [1994]</td>
<td>The New Dynamics of Global Manufacturing Site Location</td>
<td>Different geographies have different technical skills which can become strategic considerations.</td>
</tr>
<tr>
<td>Bartmess et al. [1993]</td>
<td>Building competitive advantage through a global network of competencies.</td>
<td>Well developed interfaces between plants must be represented in the global technology network.</td>
</tr>
<tr>
<td>Seitz and Peattie [2004]</td>
<td>Meeting the closed-loop challenge</td>
<td>In a globally distributed factory environment, information management is important for the entire lifecycle of the product.</td>
</tr>
<tr>
<td>Blaxill and Hout [1991]</td>
<td>The fallacy of the overhead quick fix</td>
<td>New technologies are required to build sustained knowledge transfer mechanisms.</td>
</tr>
<tr>
<td>Pisano and Wheelwright [1995]</td>
<td>The new logic of high tech R&amp;D.</td>
<td>Knowledge-based companies cannot just invest in innovative R&amp;D and outsource manufacturing – they need to invest in technology to support the manufacturing process too.</td>
</tr>
</tbody>
</table>

### Importance of Geography in Locating Internet Technology Development

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>Key Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacCormack et al. [1994]</td>
<td>The New Dynamics of Global Manufacturing Site Location</td>
<td>Different geographies have different technical skills which can become strategic considerations.</td>
</tr>
<tr>
<td>Bartmess et al. [1993]</td>
<td>Building competitive advantage through a global network of competencies.</td>
<td>Well developed interfaces between plants must be represented in the global technology network.</td>
</tr>
<tr>
<td>Seitz and Peattie [2004]</td>
<td>Meeting the closed-loop challenge</td>
<td>In a globally distributed factory environment, information management is important for the entire lifecycle of the product.</td>
</tr>
<tr>
<td>Blaxill and Hout [1991]</td>
<td>The fallacy of the overhead quick fix</td>
<td>New technologies are required to build sustained knowledge transfer mechanisms.</td>
</tr>
<tr>
<td>Pisano and Wheelwright [1995]</td>
<td>The new logic of high tech R&amp;D.</td>
<td>Knowledge-based companies cannot just invest in innovative R&amp;D and outsource manufacturing – they need to invest in technology to support the manufacturing process too.</td>
</tr>
</tbody>
</table>

### Organizations and Internet Technology

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>Key Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orlikowski and Yates [2002]</td>
<td>It’s About Time: Temporal Structuring in Organizations</td>
<td>Temporal structures are defining characteristics of organizations that need to be built into the supporting technology..</td>
</tr>
<tr>
<td>Sambamurthy et al. [2003]</td>
<td>Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms</td>
<td>Internet technology investments are indeed key influencers of firm performance.</td>
</tr>
<tr>
<td>Champy [2003]</td>
<td>Is technology delivering on its Productivity Promise?</td>
<td>Transparency, standardization, and harmonization are key technical aspects which have a positive impact on the organization.</td>
</tr>
<tr>
<td>Lacity et al. [1995]</td>
<td>IT outsourcing: maximize flexibility and control</td>
<td>Dynamic knowledge flow around the internet is key to continuous learning.</td>
</tr>
</tbody>
</table>
### Internet Security

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rivest et al. [1978]</td>
<td>A Method for Obtaining Digital Signatures and Public-Key Cryptosystems</td>
<td>Security is hugely important in transactions of sensitive information in the 24-hour KF.</td>
</tr>
<tr>
<td>Boneh, et al. [1997]</td>
<td>Fine-grained control of security capabilities</td>
<td>A fine-grained control over security privileges is required, along with the power to revoke client certification.</td>
</tr>
<tr>
<td>AES [2001]</td>
<td>Announcing the Advanced Encryption Standard</td>
<td>A FIPS-approved cryptographic algorithm can be used to protect electronic data.</td>
</tr>
<tr>
<td>Rivest et al. [1978]</td>
<td>RSA Cryptography Standard</td>
<td>The RSA Algorithm can be used for public-key cryptography with focus on the following aspects: Cryptographic primitives, Encryption schemes Signature schemes with appendix, ASN.1 syntax for representing keys and for identifying the schemes</td>
</tr>
</tbody>
</table>

### Internet Protocols

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hickman [1995]</td>
<td>The SSL Protocol</td>
<td>SSL is used to provide authentication and encrypted communication and for security-sensitive communication, such as payment transactions.</td>
</tr>
<tr>
<td>Fielding et al. [1997]</td>
<td>Hypertext Transfer Protocol -- HTTP1.1, RFC 2068</td>
<td>The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems, as mentioned by Fielding and group.</td>
</tr>
<tr>
<td>Klensin [2001]</td>
<td>Simple Mail Transfer Protocol RFC 2821</td>
<td>The SMTP protocol transports mail across networks, usually referred to as &quot;SMTP mail relaying&quot;.</td>
</tr>
</tbody>
</table>

| **Table 1. Summary of Relevant Ideas and Concepts** |
6. Technology Framework for 24-Hour Knowledge Factory

When employing a globally dispersed workforce, one needs to define how to decompose professional endeavors into a series of tasks that can be performed by each professional, without having to know most of the idiosyncrasies of other tasks. Is it practical to make this decomposition for various white-collar professions, when the concerned persons are located thousands of miles from each other? Berger [2005] describes multiple evolutions of technology that have seen the splitting of tasks as the catalyst for new work models. For example, the IBM System 360 represented the disaggregating of the mainframe hardware from the software logic, and allowed separate workers to add value to each component separately and without knowledge of the other. Similar models can be identified even within the software industry – operating systems allow software programs to be produced without knowledge of the underlying operating system; and object-oriented programming is built on the notion of separation of tasks into self-defined objects, which do not need to share their implementation with other objects in order to be used. On a broader level, the field of “component-based software” has attempted to address this need to some extent by decomposing a software endeavor into a series of components. Now the challenge is to extend this decomposition concept to various types of endeavors, both within and outside the arena of information technology.

In the 24-hour Knowledge Factory, knowledge is passed from location to location in rapid motion, and must be structured appropriately to minimize the time required for a worker to absorb the previous knowledge worker’s information. The technology used to transmit the information must be user friendly, reliable across network irregularities, and
provide a framework for the user to structure the information in the most useful manner possible. One of the key requirements of the 24-Hour Knowledge Factory paradigm is to provide new methodologies and tools that will allow an individual to understand in (literally!) 16 minutes the work done by others in the preceding 16 hours! For 24-Hour Knowledge Factories to succeed, business processes must be aligned with information systems so that patterns of business knowledge can be reused across processes [Mitra and Gupta, 2006]. In order for decisions to be made in a timely and effective manner, one needs integrated decision support systems to mitigate the spectrum of problems encountered with a geographically and temporally distributed decision making team.

When knowledge is spread over multiple locations and time zones, new IS methodologies and tools are needed to allow knowledge from anyone to be used anytime and anywhere in the process without imposing a significant burden; as such, the knowledge transfer activity becomes a critical aspect of the new IS methodology. Some preliminary tools have now become available in this arena. For example, UMEA is a tool for automatically organizing data objects into collections that are specific to tasks; it is claimed to be simpler and more productive than a file system - this type of approach reduces the time to navigate through an information repository and makes it possible to locate task specific knowledge on a daily basis as required by the 24-Hour model [Kaptelinin, 2003]. The Lifestreams concept of UMEA logs objects and events relative to a project, so that an IS engineer’s code can be automatically justified by searching the Lifestreams memory for all objects referenced by the engineer. The benefit of this approach is that prior knowledge is stored in the context of the actual artifact being worked on, so that the
worker does not need to leave the context of his or her work to retrieve knowledge related to that work [Fertig, et al., 1996]. The notion of time travel as a means of knowledge sharing is a foundational concept of Lifestreams and enables retrieval of information based on logical constraints such as 'show me the state at the time when I receive an email from 'yz' [Rekimoto, 1999]. In the 24-Hour Knowledge Factory model, the ability to introduce time and state as search parameters for knowledge is important because it allows individuals working on a rapidly-evolving project to revert to a prior day or week if they need to understand the sequence of changes that were made to the project.

When the goal is to share with others in 16 minutes what was done in 16 hours, as shown in Figure 3, there is not adequate time to fully explain how and why the decisions were made. Furthermore, when the individual is performing the work in one time zone, the opportunity to consult the colleague who made certain decisions does not exist. As such, capturing and storing the decision rationale and history in context of the information system is especially vital in the 24-Hour Knowledge Factory where workers collaborate on a project by decomposing the project into subsections that can be performed independently and then merged together on a continual daily basis. Some of the previous projects in this arena have been domain-specific. For civil engineering endeavors, the DICE (Distributed and Integrated Collaborative Engineering Environment) methodology provided a platform for collaborative engineering; it decomposed each engineering project into a set of modules and allowed work to be conducted in parallel on each section of the project [Sriram, et al., 1992]. When the system encountered conflicting decisions about a particular design decision from engineers in different modules, it used
the design rationale to help negotiate the outcome. The use of such a strategy in the 24-Hour Knowledge Factory would enable workers, especially ones with conflicting solutions to a given problem, to gauge the exact implications of each option on the different stakeholders of the system.

**Process Timing**

![Figure 3: Work done in 16 hours must be “handed off” in 16 minutes, literally!](image)

The existing field of design rationale capture tools spans the spectrum from fully unstructured rationale to completely modeled rationale. Minutes of meetings represent an unstructured, time delineated capture of information. QuestMap [2000] and DRAMA [Brice and Johns, 1998] are examples of the next step – they provide basic structural elements and enable the user to devise a useful structure. At the other end of the spectrum from meeting minutes is DRIM, a system that incorporates a completely specified model for the rationale underlying the design process [Pena-Mora, et al., 1995]. Design reuse is an obvious requirement in the 24-Hour Knowledge Factory – the ability to componentize and plug in work done in previous time periods enables the process to progress much faster. The SPOOL project at the Université de Montréal uses reverse engineering techniques to analyze existing projects and to determine which patterns in the
design could be used to infer the rationale behind recurring patterns [Keller, et al., 1999]. This minimizes the need for the designer to provide the rationale so that the 8-hour period in the 24-Hour model can be spent on producing the core information, rather than on documenting the rationale.

Based on the above considerations, two prototype systems were conceived. In the first prototype, an automated system for capturing design rationale and history was made possible by having a web-based form and database available to stakeholders across the internet. In the second prototype, the notion of composite personae allowed multiple individuals to act as the same individual when interacting with the knowledge-based system via the internet. In both cases, internet technology was the key enabler for creating the new framework for transferring knowledge: an essential prerequisite of the 24-hour knowledge factory.

**KNOWFACT: Internet-based tool for Capturing Design Rationale and History**

Under the aegis of a research contract from an agency of the US Government to MIT, a concept demonstration prototype, KNOWFACT (from Knowledge Factory), was designed, developed, and implemented for sharing structured knowledge across the internet. The efficacy of this prototype was evaluated in a real-world situation involving the design of satellites, where technical decisions regarding the specific design of the system are made on a daily basis, and require constant re-evaluation of the effects of these decisions on a variety of stakeholders who are geographically distributed. The stakeholders used an internet-based system for providing structured insights into the
design of the system, without requiring them to interact with the designers or the other stakeholders on a synchronous basis.

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision History Module</td>
<td>Design Rationale</td>
<td>Design Parameters</td>
<td>Attribute Values</td>
</tr>
<tr>
<td>Decision Rationale Module</td>
<td>Attribute Definition</td>
<td>Utility Interview</td>
<td>Utility Function</td>
</tr>
</tbody>
</table>

KNOWFACT comprises of two modules, as depicted in Figure 4. The Decision Rationale Module (DRM) is used to create the definition of the key attributes of the system, based on inputs provided by individual stakeholder via a web-based interview form. Collectively, these attributes constitute the framework for a utility interview that helps to analyze the extent to which the system caters to the requirements of the stakeholders.

The Decision History Module (DHM) captures historical information on specific decision parameters, in incremental form, via a web based interface. Based on the response of the stakeholder to a particular question, the next question is created. This minimizes the incidence of irrelevant and inappropriate questions. Dynamic forms are used for human interaction with the system, based on earlier research that decision support systems with flexible forms that are appropriate to the data being used for decision-making are most useful to decision-makers [Wu, et al. 2004]. DHM provides an evolving repository that contains knowledge on the current state of the activity being undertaken; it also performs an automatic analysis of system integration integrity, with the objective of alerting the integrator of potential conflicts.
DHM captures and reuses valuable information assets, with the objective of mitigating temporal and spatial barriers in large multi-organizational multidisciplinary endeavors. For diverse applications, ranging from creating marketing campaigns to the design of new systems, each endeavor is conducted as a new one, with zero or minimal knowledge being carried over from the previous endeavor of very similar nature. With the objective of mitigating this problem and facilitating use of historical information, DHM was designed to automatically capture and store important information (name, value, rationale, author, etc) relating to each component of the project, along with the history of these data. In addition to the automated capability to capture vital information, DHM is designed to prompt the user to document the rationale when significant changes. This allows individuals to look readily at decisions made on earlier projects that involved decisions on similar issues. In the context of the 24-hour knowledge factory paradigm, this approach will enable knowledge to be shared between teams working in different places around the globe, with minimal human overhead involved in transferring the knowledge of previous projects.

The Decision Rationale Module (DRM) elicits and captures critical information on the objective of the endeavor, and then analyzes such information for every stakeholder. Prior research that suggests that quality of team-based decisions is optimal when the decision can be described in terms of multi-attribute models. DRM allows stakeholders to interact at the attribute level. The specific parameters involved in the decision are related to the attributes once and once only; then the utility measures draw exact links
between the parameter changes and the effect on the overall attributes. Utility analysis methods and cost-benefit analysis methods are used to elicit information from all stakeholders in the geographically and organizationally decentralized design process in order to facilitate communication between them. These data are analyzed to ultimately provide information on the utility and the cost of each potential architectural alternative. This approach enables opinions of the decision-makers’ to be accorded equitable representation, even in cases involving major temporal and spatial separations, such as in the case of the 24-hour knowledge factory environment.

The system was designed to incorporate only a small set of aggregate attributes because it has been shown that decomposing a system to a small set of aggregate attributes has a positive effect on human comprehensibility and accuracy in terms of interacting with a decision support system [Bohanec and Zupan, 2004].

**MULTIMIND: Internet-based Prototype for Sharing Tasks with Composite Personae**

Many disciplines, including software engineering recommend the approach of “divide and conquer,” that is, decompose the main problem on a recursive basis into increasingly smaller sub-problems and subtasks until the stage the “component” can be addressed. In modern parlance, the result of such decomposition for a software application is a set of modules and classes (expressed in an object-oriented language such as Smalltalk, C++, or Java). Each module or component is next assigned to a developer who carries it through the process of coding, testing, and possibly even maintenance. In the traditional
paradigm, the “ownership” rights to each component were typically vested in only one person, who was responsible for ensuring that the actual performance of the component matched its expected state, and for overcoming defects and inconsistencies. While multiple programmers participate in the coding process, the owner of the component bears ultimate responsibility for bug fixes and possesses the most comprehensive knowledgeable of that particular component. With sites that have zero overlap in work schedules, as in the case of the 24-Hour Knowledge Factory, the single ownership model is obviously inappropriate.

Based on the assessment that the lightweight, self-optimizing Personal Software Process (PSP) offers an attractive platform for agile software development in the 24-Hour Knowledge Factory, and the realization that PSP assumes that artifacts have single ownership, the concept of composite persona (CP) was formulated. A CP is envisaged as a highly cohesive micro-team that, like a corporation, possesses simultaneous properties of both an individual and a collection of individuals. A composite persona is designed to act as a singular entity, even though it comprises of (at least) three individuals, with one person at each site, with one site being active at any point in time. This configuration does not imply, however, that all sites possess the same number of persons; this is because one developer may belong to more than one CP. Using this notion of CPs, development can proceed in a manner similar to the traditional one, with the sole difference that the owner of each component is not an individual, but a CP. This CP is actively involved in the process of development and conflict resolution on a round-the-clock basis.
Figure 5 shows a situation involving interaction between two composite personae: CP Mercury and CP Athena. The arcs between members of each CP illustrate the flow of information/knowledge from Australia, to Poland, to the US, and then back again to Australia. Note that when the site in Poland is on, Gzegorz, the driver of CP Mercury, and Sylwester, the driver of CP Athena, can communicate conversationally in real-time. Thus, both asynchronous communication within the CP and synchronous communication between CPs occur over time, based on need.

Since communication between CPs is essentially one-way, from one driver to the next, one must address the possibility that questions remain after the handoff. As the new driver cannot query the previous driver directly, a new approach for code justification was conceived by Nathan Denny, Shivram Mani, and other members of the project team at the University of Arizona. In this approach, the development system tracks all...
The fundamental operation in the context of CPs is the handoff. The handoff procedure used during the implementation of the concept development prototype system is derived from the one used in the Scrum agile process. In the latter case, Scrum stand-up meetings occur for co-located developers, first thing in the morning. All attendees briefly summarize what they accomplished the previous workday, what problems they encountered, and what they expect to accomplish during the particular day. In our case, such debriefing takes place via electronic means. Our approach is somewhat similar to UMEA [Kaptelinin, 2003] in other respects.

The approach used for creating the project’s knowledge base is based on the Lifestreams concept [Fertig, et al., 1996]. Relevant objects and events are posted and logged into a monotonically increasing database that is ordered on a chronological basis. This capability is central to justify decisions made by one user to a different user that is trying to understand why a particular decision was made. For example, a block of code can be justified to a reader by searching the knowledge base for all relevant objects consumed by the author of the code block just prior to the creation of the block.

Multimind, is a stand-alone system for facilitating the execution of vertically decomposed tasks by distributed teams, in a sequential manner. In the design of this prototype, the objectives were: automate what can be automated; store everything that can be stored;
and, do not overwhelm the decision maker with trivial, irrelevant, or redundant information. The system caters to the following functionality:

- defines atomic editing operations;
- stores every edit (down to the atomic level), every communication (e-mail, chat, etc), every document browsed/read, and every search made;
- creates summaries automatically by monitoring activity of each worker and reporting in a hierarchical fashion where and what changes have been made;
- creates informational summaries by compressing the edit log.

The above framework is used to automatically create work summaries for each shift. The template-based scrum-style system is also designed to allow each worker, on clock-out, to make very quick observations on the anticipated future state of the concerned sub-tasks to assist the fellow worker who would be taking over the responsibility for continuing the concerned pieces of work. Since the next worker must understand the state of the task, this is a decision support system in reverse, that is, a decision justification system. To accomplish this objective, the integrated project knowledge base is mined for relevant knowledge objects (chats, e-mails, documents, etc) that relate to the work done by the concerned worker in the previous shift. The objective is to provide the incoming worker with only relevant and condensed knowledge objects.

We distinguish between project objects and events. Project objects are objects that evolve as the project progresses; this includes: list of requirements, source code, UML models,
and object code. Events are significant worker initiated actions that carry useful information on what the worker was doing at a particular time. We track the following events with an associated timestamp:

- **LoginEvent**: registered when an agent begins his/her work/shift;
- **LogoutEvent**: registered when an agent completes his/her work/shift;
- **MessageViewEvent**: registered when an agent opens a message for reading;
- **MessageSearchEvent**: registered when an agent issues a search in the content of his/her messages;
- **ExternalWebAccessEvent**: registered when an agent loads a webpage in his/her browser;
- **ExternalWebSearchEvent**: registered when an agent issues a web search (e.g. a query at google);
- **DocumentSearchEvent**: registered when an agent issues a search in the corporate memory for a document (such as design documents and test documents).

When a person wants to view the rationale behind the state change of the project object, the knowledge base can be queried for all the events based on the timeframe when the state evolved. The results are categorized based on the type of event, and prioritized and ordered based on relevancy.

Based on our experience that computer professionals do not always enter the rational on the computer, we are augmenting the prototype to cater to voice inputs. This will allow the programmers to annotate their code more easily. The voice inputs will be translated into text form. One will therefore have the facility to click on a piece of code, and to
know the rationale for it either in audio form or in text form. Each comment is linked directly to the appropriate place in the code.

Further, the notion of CP is being reinforced with the notion of the Composite Personae Software Process (CPSP). Based on the concept of Personal Software Process, CPSP attempts to enhance the productivity of the CP by providing automated mechanisms to estimate delivery schedules, reduce defects, propagate knowledge within the CP, and assign workloads to members in a manner that maximizes overall productivity. When the class or module is first assigned to the CP, each member of the CP makes an estimate of the complexity of each phase of each task. These estimates are used to decide, via a Monte Carlo simulation approach, how the overall task should be performed (see figure 6 below). Further, since the participating teams are located in geographically different locations, a decentralized approach, involving peer-to-peer concepts, is being analyzed in terms of performance and latency considerations.

Figure 6: Dynamic Reassignment as Requirements Change
7. Examples of Harbingers of 24-Hour Knowledge Factory

The use of multiple collaborating centers spread across multiple continents is becoming prevalent at several computer companies. A controlled experiment was conducted, at IBM, to compare the performance of a team working exclusively at one place and another team that involved workers at one location in the US and a second location in India. The two software development teams studied at IBM were virtually identical in all structural respects. Each team had seven core developers, of similarly varying experience and responsibility; they were organized on a task-oriented basis, in terms of the above hierarchy. Both teams were managed by the same development manager, and were observed closely for a period of 52 weeks.

The results of this controlled experiment are interesting. In the co-located scenario, knowledge was held only at one location; but in the distributed scenario, with tasks being interdependent and shared between locations on a regular basis, knowledge was disseminated as a natural part of the process, thereby leading to diversification of knowledge resources. Further, the time taken for resolution of tasks was reduced by nearly 50% in the latter case. There was an unintended process improvement: with the introduction of the 24-hour development model, the capture of the decision rationale and history became a natural part of the development process, and the tendency of software engineers to avoid or delay knowledge dissemination tasks was overcome. On the flip side, the loss of informal communication was cited as the most significant hurdle. In order to partially mitigate this problem, all the members of the distributed team had an initial face-to-face meeting.
Motorola is another company that has utilized a similar approach to achieve higher levels of productivity from their software division. A project at Motorola used developers in six countries to develop the same piece of software, with each site maintaining local work shifts. Work was performed on the project, somewhere in the world, for at least 21.5 hours per day [Battin, et al., 2001]. Over 500,000 lines of code were developed with minimal synchronous interaction among the different sites.

Fujitsu has employed an object-oriented team approach to achieve 24/7 support of their technical systems with a combination of local teams, continental teams, and global teams to handle support issues of mission-critical systems on a 24/7 basis. Certain issues are handled locally, certain ones are passed between offices, and certain ones are global and so knowledge is stored centrally. Decisions are made on a case-by-case basis on how serious an issue is and how broadly to escalate it.

On the manufacturing side, Siemens AG has automated tools that can be monitored from any geographical location, and handled remotely. In this case, the traditional factory environment has been transformed to a virtual “knowledge factory’ where operation continues without having any factory workers have to work the night shift. Motorola developed its Bandit line of pagers with a “heavyweight team” that took responsibility for the substance of its work, the means by which it was accomplished, and its results. The project was completed in 18 months (which represented about half the time of a normal project of such magnitude).
Yap [2005] describes a globally distributed, round-the-clock software development project involving a programming team, distributed across three sites (US, UK, and Asia), that used collective ownership of code. One of the three sites had prior knowledge of extreme programming. The two remaining sites were coached on extreme programming practices ahead of the collaboration endeavor. These two sites believed that the first site had an advantage due to its previous knowledge with extreme programming. Persons from the three sites also met in person, which they felt helped the program start by building confidence in the members of other sites. The team used Virtual Network Computing (VNC) and video conferencing to facilitate communication. Handoff of project artifacts initially consisted of a daily work summary, but later grew to include knowledge learned and new objectives. This three-site system encountered problems in the form of confusion caused by cultural differences and increased wait time due to downloading through their version control software from distant locations. They learned that each location needed an equal sized team to prevent the largest team from taking over the design. They also realized that priorities were changing too quickly so they decided that managers could only reprioritize things once a week.

The current set of evolving applications can be broadly grouped into two major categories. The first pertains to the design, development, and implementation of information systems in a manner that leverages the new paradigm. The use of the 24-Hour Knowledge Factory paradigm can allow each of these tasks to be performed at a faster pace and at a lower cost. In today’s parlance, the 24-hour Knowledge Factory will
involve “partial offshoring” or “hybrid offshoring” as an integral part of the endeavor, with one team being deemed as onshore and the other two teams as offshore. This distinction between onshore and offshore is primarily relevant at the time of initiation of a project – over time, the 24-Hour Knowledge Factory model seeks to establish a seamless integration that enables each site to be an equal contributor. Examples of this can be seen in the multi-shore operations of firms such as Pipal Research Inc. [Wharton, 2006].

The second involves the application of the 24-Hour Knowledge Factory paradigm to a broad range of other wide collar activities ranging from medical services to logistics planning, and from financial analysis to product design. Consider the functioning of the firm, OfficeTiger Inc. based in the US, India, Sri Lanka, Switzerland and the Philippines [Wharton, 2005]. One-third of their deadlines are shorter than three hours, and about half of their deadlines relate to work that needs to be completed within a day. In cases such as OfficeTiger, the challenge is to design, develop, and deploy the infrastructure that can be utilized by these industries in order to perform their respective endeavors at lower costs and with shorter turnaround times. Ideally, one would like to use the same infrastructure to provide 24-Hour Knowledge Factory capabilities to diverse arenas; in reality, the infrastructure may need to be adapted to address the specific requirements of different endeavors.
8. Continuing Tasks and Challenges

Prior to the industrial revolution, a person would be engaged in developing an item from start to finish. If a hammer had to be fabricated, the person would hold responsibility for cutting the wood, trimming the wood, polishing it; the concerned person would also be responsible for heating the metal and bending it to the right shape. In every sense of the word, the end product was a piece of art, manifesting the values of the artisan who created it from scratch. The advent of machines changed the entire operation. Multiple persons, in fact sets of persons spread over two or three shifts, would perform micro-tasks on each item. The items themselves got standardized over time, thereby losing the customization aspect.

While the Industrial Revolution discarded the notion of specialized artisans in favor of assembly lines producing generic goods; the evolving Knowledge Factories can bring the artisan back to center stage, emphasizing customization over generalization, and individuality over group behavior. The new set of services can be tailored to reflect the individual skills of the worker, and to use them to create customized products and services. For example, one can now use the Internet to order a custom bicycle be manufactured to individual specifications and be truly one-of-a-kind. Broadening this example, we can envision a new generation of service workers who will help to design, manufacture, and market customized products utilizing the most cost-effective options at each stage. As one begins to access and use less expensive labor in places around the world, one can create custom products and services at lower costs.
Further, in a world characterized by growing amounts of outsourcing, the new paradigm provides much greater potential for individuals to progress in their respective fields of expertise. Workers can gradually move up the value-chain and provide a growing number of higher-value services.

Also, conventional factories employed workers from very similar backgrounds and national origins. In the 24-hour decentralized development process, professionals from different cultures are concurrently engaged on all tasks, and can provide their cultural input on a continuing basis into all tasks at all stages of the process. This paradigm works better than existing mechanisms for understanding requirements from different cultures, thereby producing end services and products that command greater appeal in the global economy.

In order to address the above types of issues, we still need to find efficient ways for decomposing diverse types of professional endeavors into a series of tasks that can be readily transmitted over the internet and be capable of being performed by a professional, without having to know most of the idiosyncrasies of other tasks. While the concept of decomposing major endeavors into a series of discrete tasks has been embraced by part of the information systems communities and by sections of few other communities, we need to broaden this vision to include many diverse types of professional activities.

In order to surmount spatial, temporal, and other kinds of boundaries, Gupta [2001] advocated a four-faceted knowledge-based approach that placed emphasis on:
Knowledge Acquisition or tapping traditional systems to provide accurate and comprehensive material for the new knowledge-based systems;

Knowledge Discovery or automated mining of numerical, textual, pictorial, audio, video, and other types of information, to capture underlying knowledge, either on a one-time basis or on a continuous basis;

Knowledge Management to deal with different types of heterogeneities that invariably exist when inputs have to cross-over borders of different types (national, organizational, departmental, and other); and

Knowledge Dissemination to extract, customize, and direct knowledge to appropriate departments and users, based on their individual needs.

The current versions of Internet are geared primarily to provide physical connectivity. As the internet evolves, it will incorporate additional layers for providing logical connectivity, thereby facilitating the agile creation and deployment of 24-hour Knowledge Factory capabilities.

9. Conclusions

The 24-hour knowledge factory utilizes multiple sets of workers, based in three or more different continents, to contribute their talent in a sequential manner to create the appearance of non-stop work. This new concept, akin to the concept of shifts that originated as a vital component of the industrial revolution, can significantly reduce development costs and development times for a number of knowledge-based endeavors.
in the area of information technology. In addition, it offers the potential for creating the information technology infrastructure that can be utilized in other white-collar professions. The 24-hour knowledge factory has been made possible by advances in internet technology; in turn, the use of the 24-hour knowledge factory concepts will play a significant role in the development of next-generation internet technologies. This paper has focused on the modeling and implementation of a prototype system that reflects some of the core characteristics of the 24-hour knowledge factory model. As this globally distributed work paradigm evolves, other industries will adopt and benefit from the 24-hour knowledge factory model.

10. Acknowledgements

The authors thank a number of individuals at the University of Arizona, MIT, IBM, and in other organizations for their valuable ideas and for providing information on their specific endeavors. In particular, the authors thank Nathan Denny, Vinayak Sambargi, Shivram Mani, and Srividhya Subramanian for their insights on Composite Personae and underlying Internet technologies. At various stages of its evolution, this paper was reviewed by several distinguished domain experts; the authors are sincerely grateful to them for their frank and constructive feedback.


Hickman, K., Feb 9, 1995. The SSL Protocol, Netscape Communications Corp.


Klensin J. Apr. 2001 Simple mail transfer protocol. RFC 2821.


Wharton 2006: Realizing the Indian Dream. 11th Wharton India Economic forum.
