Creating an Integrated Platform for Enterprise-Wide Process Intelligence

Roy Altman, Peopleserv, Inc., USA

INTRODUCTION—WHAT IS YOUR STRATEGY?

When we talk about a company's strategy, what do we mean? How is it different than the operational objectives of maximizing income, reducing costs and risks? Each company must decide for itself what its strategy is, and what business outcomes to tie it to. The role of technology should be to assist a company in achieving that strategy.

Many companies confuse strategy with tactics. They think that stringing together tactics, although worthwhile objectives by themselves, will amount to a strategy. It will not. An example would be an HRIT department wanting to deploy learning management, talent management, talent acquisition and workforce management solutions as its strategy. Each of these solutions may be part of an overall strategy to attract, retain and develop talent for the organization, but these goals should be a stated objective that helps define the tactics for achieving those ends.

Earlier in the history of the information technology field, having state-of-the-art business technology was sufficient as competitive differentiator, because the field was new, technological standards were still in development, and techniques to effectively manage the deployment of technology were not mature. In 2003, Nicolas Carr published a famous paper in *Harvard Business Review* entitled: "Does IT Matter,"[1] in which he argued that technology had become commoditized. In his view, having superior technology did not truly deliver a competitive advantage. The best way to manage a commodity was to reduce costs while retaining reliability. This paper resulted in an uproar among the IT community (which had a vested interest in maintaining its status in the organization). Entire books were written to refute it, including Howard Smith and Peter Fingar's "IT Doesn't Matter, Business Processes Do."[2] Smith and Fingar's point was: it's not the technology per se that creates a competitive advantage; it's how you use it to solve problems consistent with the business outcomes you're trying to achieve.

Business Process Management focuses on the end-to-end business processes and attempts to optimize them using technology. Regardless, whether interacting software components or humans serve the processes, it's the end-to-end process that counts, not the snippet of the overall process.

MOVEMENT TO THE CLOUD

There is a trend in IT to move applications to the cloud. The advent of the World Wide Web meant that when application software had to be upgraded, it no longer needed to be distributed to each desktop. If the system ran in a browser, the update occurred at the server and it would automatically affect all users. However, web architectures existed in-house, behind the company's firewall. This was a holdover from the client-server days (actually much "web" based software was originally client-server retrofitted to the web once that technology became available) and reflected customers' purchasing preferences. However, with the advent of strong encryption technology, which became legal to export in the late 1990's, there was no technological need to house servers at the company's site and still maintain secure storage and transmission of information. As a matter of fact, business-to-consumer (B2C) applications had already been working that way for several years. The trend began with application providers "hosting" applications for clients in their own data centers. Therefore it was the responsibility of the application provider to maintain the hardware and software, and apply updates as they become available. The business model changed as well. Instead of purchasing a perpetual license and paying ongoing maintenance fees, companies would just pay a monthly fee for usage of the software. This trend became known as Software-as-a-Service (SaaS).

SaaS trend was at first met with resistance from companies that were reluctant to give up control over where their information is stored. Over time, though, companies have become increasingly accepting, as vendors have been able to demonstrate their (often superior) security measures, along with the many advantages of SaaS:

- Predictable costs
- Funding comes from the company's operational, not capital budget
- No need to maintain IT staff to maintain hardware needed to run the application
- No need to devote resources, time and budget to applying upgrades when available
- The vendor is responsible for keeping the software current for compliance purposes and is able to deliver accelerated innovation to the customer

Traditionally, companies would license software, and then customize it for their unique business processes. This practice made upgrading the software more difficult because the customizations needed to be reapplied to the new version, which could be problematic if architectural changes were required as part of the upgrade that conflicted with the design of the customization. There is an implementation of SaaS many call "pure SaaS" which means that all customers' information resides in the same, multi-tenant database, and all customers are executing the same version of the application. This means that the application can only be configured, not customized, to the business needs of each customer. This has the advantage to the vendor that the environment of all clients is the same so upgrades can be applied easier and thus more frequently.

The advantage to the customer is that it can receive updates several times a year, instead of once every few years, thereby enjoying software that is continuously innovated and improved. However, it means that the design of the application must be flexible enough to be configured for the business needs of every customer without having to be customized. Customers may inevitably have to make some compromises, but on balance it is worth the trade-off.

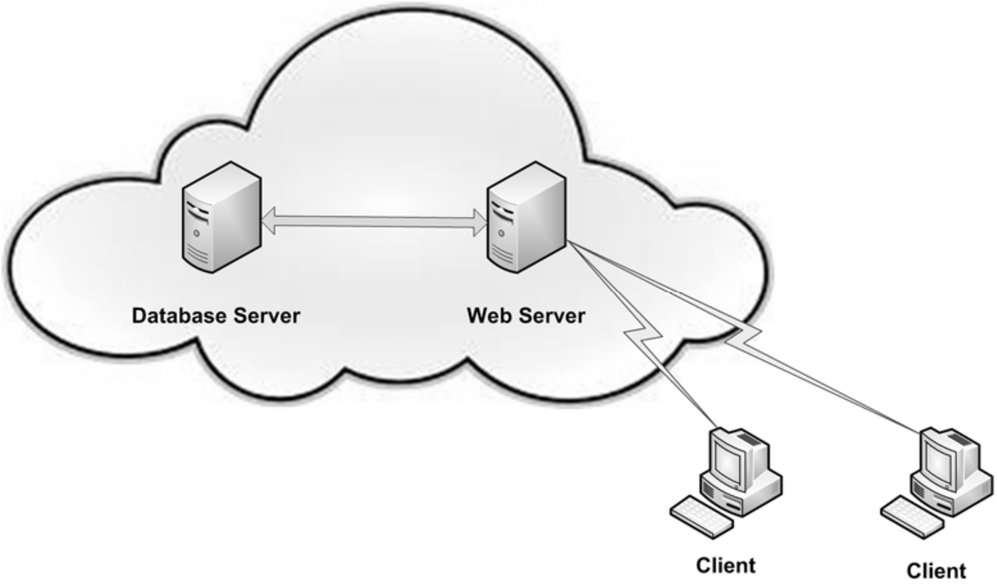

Fig. 1: Cloud-based architecture of today

CHANGING ROLE OF IT

With the acceptance of cloud-software in the organization, the role of IT is changing. IT is no longer responsible for maintaining computer hardware, internal networks, and software upgrades and tuning. So what should IT do? Was Nicolas Carr right; are IT's days of contributing strategic, competitive advantage over? IT can now focus of getting closer to the businesses they serve. It is IT's role to integrate the many systems, both in-house and cloud-based, to ensure that each software asset is available to all users who need them, and that software agents operate as an integrated whole, rather than silo'd horizontal towers. Thus IT should focus on facilitating end-to-end business processes for the user, leveraging any software agent that fulfills any part of the business process. IT's role, therefore, rather than being minimized, is being elevated to a more strategic function, if CIO's are savvy enough to take advantage of this opportunity.

INTEGRATIONS

Integrations have always been both the bane and Holy Grail of IT. Since the dawn of the digital era, it has been recognized that systems were designed to do a specific thing and needed to interface with other systems to accomplish business processes. With the emergence of the World Wide Web (WWW) in the 1990s, the stakes became higher as business-to-business (B2B) e-commerce depended on disparate systems interacting with one another over a public network. Before the Web age businesses have made many attempts at integrations, although on private networks.

History

Flat file interfaces have always existed. The problem with them is that the sending system needs to know the format of the file that the receiving system expects. If the file format is at all different, the interface will break, revealing its fragility. A commonly used flat file format is comma-separated values (CSV), which is supported by Excel spreadsheets. It's shocking to note that this is still the most commonly used file format for interfaces between systems.

Also, flat file interfaces must be run in batch mode, meaning that the interface cannot occur in real time, but on a pre-set schedule. Many processes would benefit from real-time interfaces.

Early attempts at automated interfaces yielded a standard called the Common Object Request Broker Architecture (CORBA) released in 1991. Designed by a committee of the Object Management Group (OMG), CORBA was considered by some complicated to implement and never reached widespread acceptance. Microsoft released its own standard, called Common Object Model (COM) in 1993, which suffered the same fate.

Companies involved in e-commerce needed a standard for computer-to-computer interchange to execute the transactions. To that end, Electronic Data Interchange (EDI) was released by the National Institute of Standards and Technology in 1996. A data standard was developed for each industry, thus it represented a tight coupling that lacked flexibility.

Current technology

Enter the age of the World Wide Web. In his 1999 book "Weaving the Web,"[3] original designer of the Web Tim Berners-Lee likened adoption to a bobsled, which starts slowly and is very soon hurtling at tremendous speed. The Web is based on Hyper-Text Markup Language (HTML), which is a standard markup language that describes how a document should *look*. The Web quickly became pervasive because HTML is a based on a single standard that is adhered to by everyone. This is in stark contrast to the previous generations, when industry "standards" became de-facto, perpetuated by the perceived market leader of the time.

Berners-Lee felt no one should own web standards, so he formed the World Wide Web Consortium (W3C) to determine universal web protocols.

In 1998, the W3C proposed a standard called Extensible Markup Language (XML) that describes what data should *mean*. An XML document has tags indicating what the data item is along with the value. This method of data transfer is not fragile as flat files are, because an XML parser can search for targeted data items regardless where they reside in the XML document. XML has become the standard method of data interchange between systems.

The Service-Oriented Architecture

This and other protocols based on universally accepted standards spawned a software design methodology called Service-Oriented Architectures (SOA). The premise is rather than access entire software applications, parts of the applications could be accessed individually as services. For instance, in an onboarding application, a service can be called to add the employee to an external vendor who processes benefits for the company. This is an example of loose-coupling, because the calling and called service don't need to know very much about each other, just the application programming interface (API) needed to retrieve the desired information. A set of related technologies emerged which collectively comprise what is called Web Services, which are the technical underpinnings of SOA.

With the emergence of standards-based protocols supporting SOA, one would think that the Holy Grail has finally been obtained. Unfortunately, that has not yet proven to be the case. The primary obstacles to widespread SOA adoption include:

- A lack of maturity in using SOA techniques in companies, even though the technologies are mature.
- While simpler than previous generations of integration technologies, SOA integrations are still difficult and require technical expertise.
- Functionality exposed as web services, upon which integrations with divergent software depend. This is counter to the marketing strategies of many software companies, which license more software by offering a wider breadth of functionality under their own brand names, rather than encourage integration.

It is incumbent on the customers to force vendors to expose their functions as web services, so they can be integrated. Many companies shy away from integrating because of the difficulty, and vendors aren't helping this cause. As a result, we have the silo'd applications mentioned in the first section. But with the changing role of IT, perhaps we'll see progress in this area.

Levels of integration

When one says "integrations" it means many things. There are several levels of integration, ranging from simple interfaces to true interoperability. Here is some explanation:

- **Data**—each system shares a common set of information so each knows whom the current user is. Key indicative data is transferred from the system of record to every other system. For example, if you're a manager working on an employee's performance review, then you want to review her salary history; when you transfer over to the compensation function it will take you directly to the information for that employee, because the system knows who the user is and which employee she's working on. The processes are attached by means of a "deep link," which means that the link will take the user directly to the function requested (in this case the salary history page) rather than returning to the landing page of the compensation module and forcing her to navigate to the salary history page. Implicit in this process is single-sign-on, which means that the user just authenticates once and those authentication credentials are passed to any other software agent she interacts with.
- **Process**—rather than deep links, which takes one to a particular page within a module, process integration invokes a service within the module, using web-services technology described earlier. An example would be the manager is doing a performance review of an employee, and the performance module fetches all interactions the employee had with others from the company's collaboration software.
- **People**—this layer of integration involves the system knowing the context of the function being performed, and being sensitive to the important relationships the employee has with others with regard to that function. For instance, in the performance review, the system is aware of all other team members the employee works with on various project teams in a matrix-management environment. In the compensation function, the system knows to whom to route transactions for approval in the case that the recommended salary increase exceeds guidelines based on the employee's position.

A deep integration involving all three layers can be said to be a *"functional"* integration because it has contextual sensitivity to the information, the processes and the relationships between the participants.

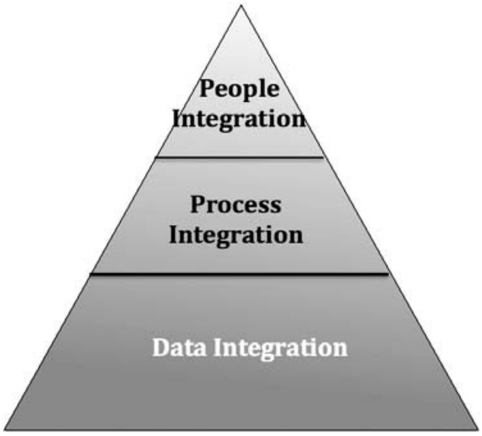

Fig. 2: Levels of integration

Today and Onward

BPM technology is effective for automating structured processes, where all of the possible execution paths are known in advance. The more recent concept of Adaptive Case Management (ACM), or Dynamic Case Management, is concerned with providing tools to empower the knowledge worker, operating in an unpredictable environment. As Paul Harmon noted in his keynote at the 2013 bpmNEXT conference[4], BPM technology evolved from early attempts at Artificial Intelligence and Expert Systems, which essentially consisted of business rules. Artificial Intelligence fell out of favor after early enthusiasm waned, but the technology has advanced and is now quite pervasive, as MS Word reminds me every time I write a sentence with an invalid structure.

All of the above technologies should be available to business users to construct an ecosystem of automated agents to assist with their work. Also at bpmNEXT, Dominic Greenwood said that BPM and ACM exist in a continuum, whereby tasks may start as structured, take an unpredictable turn, then resume a predictable path. The technologies to address each step should be readily available to invoke as needed.

The market for BPM suites and application-specific software are very competitive, with innovations frequently emerging. A customer should not be locked in to one product if another one that better suits its business needs is available. The cloud makes it (in some ways) easier and quicker to deploy applications. Users should be able to swap out software and replace it as needed.

Technology is advancing rapidly, with enhanced machine intelligence, miniaturization and new input devices. The platform should be agnostic to the point where it can integrate anything as long as it conforms to accepted standards.

THE INTELLIGENT BPM PLATFORM

At long last we are within reach of the "Holiest of Grails," a platform with true *functional* integration and interoperability. The idea behind the platform would be to "plug in" any application and have it be interoperable with any other application already connected to the platform. Whether it's a data or process integration depends on the degree to which the application exposes its processes as web services. However all applications will be "people" integrated by virtue of the centralized repository of rules, roles and relationships (see next section for more detail). Such a platform would have to possess the following characteristics:

- **Portal approach**—the platform should appear to the user as a portal, with all of the information and services aggregated on one, configurable page. The portal should know who the user is, what she is allowed to access, and what's she's likely to want to access.
- **Single sign-on**—it seems obvious, but an essential ingredient of a streamlined user experience is single sign-on. It's very annoying for a user to be challenged for login credentials each time she encounters a new system.
- **Configurable look and feel**—when switching between systems, it's best to have a consistent look and feel. To some degree this will be a compromise as the degree of configurability varies widely between software products. However, there should at least be consistent branding.
- **A centralized information hub**—see the next section for a more detailed explanation.
- **Fully web-services enabled**—should support all current accepted standards for integrations. Should be backward compatible to support several flat-file formats as well.
- **Centralized workflow**—all action items/approvals should appear on a single action list. See the next section for more detail.
- **Extended Relationship Management (xRM)**[5]—an essential component of this platform is the xRM application, which allows configuration and management of all roles and relationships in the extended enterprise.
- **Configurable by business users**—some degree of configuration should be available to business users, rather than technical folks.
- **Extensible**—the platform should not be confined to information only within the company. The value chain extends beyond the firewall, and the platform should have the ability to extend to the web for additional information.
- **True Platform-as-a-Service (PaaS)—**the platform should be a true Platform-as-a-Service. It should exist in the cloud, rather than on-premise, be a single code-base for all users, and have a multi-tenant database. Therefore, it should be highly configurable, but lack a facility for code customization (see the discussion below on extensibility, which will allow certain users to invoke business rules unique to them without infringing on the common code-base). This will allow the vendor to roll out upgrades on a frequent schedule and maintain a quick pace for new and more innovative features and functionalities, which also serves the customer.
- **Integrated analytics**—in that this is an information repository, a great deal of mission-critical information is at the user's fingertips. The platform should have an integrated analytics engine, which can provide a management dashboard of relevant metrics for the user.

- **Collaboration tools—**social media tools, such as wikis, blogs and discussion groups should be available to communities defined and maintained by the relationship management tool.

A Look Under the Hood

At the heart of the platform is the centralized information repository. It draws information from multiple systems of record, and indicates the source of each piece of information so it cannot be changed in the repository, unless it has been created there. The repository can also serve as a real-time Global Data Warehouse, and can provide analytics to the dashboard.

The Extended Relationship Management engine ensures that any time a data element changes, it's re-sent to the repository in real-time through web services. The information acts as meta-data, and can be used to construct business rules that drive groupings of workers. For instance, logical statements can be created using the meta-data (e.g.: DEPARTMENT = '26001'), which will enforce that all workers in that department will be grouped together.

Business rules can be used for any purpose, but are intended to group workers in the construction of organizational structures. These structures are the life-blood of the company, and can be mapped to any process in the company where sensitivity to organizational relationships are value enhancing. If one were to map all of the relationships that make a company work, it would be unfathomably complex. The Extended Relationship Management tool allows that complexity to be managed effectively; the meta-data driven business rules contribute to keeping the information current automatically.

Extensibility

The system is extensible in two respects: "virtual" data items and "virtual" business rules. Virtual data is information that isn't derived from internal systems through an application interface; they can exist anywhere on the web and be accessed through web services. For instance, if I want to group all stocks whose stock price is over 100, I can access the ever-changing stock prices on the web and dynamically maintain that group of stocks. This aspect of the architecture can be open source. Since the architecture has a published API, anyone can develop connectors to the information they want and make that connector available to anyone else who would like to use it.

The other aspect is "virtual" business rules. As a pure PaaS, any features designed into the product are available to all, and typically if a feature is not designed into the product it is unavailable. However, due to the published API, any user of the system can write her own business rules and host them elsewhere. The results of the business rules can be used within other business rules or to trigger behaviors within the system.

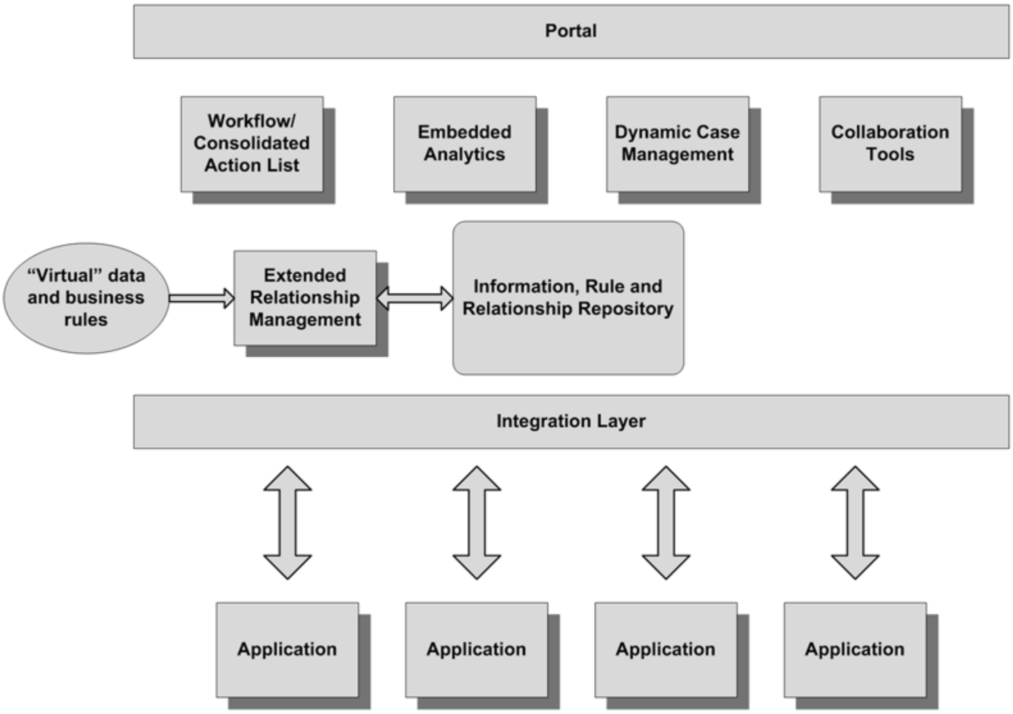
Portal
Workflow/ Consolidated Action List
Embedded Analytics
Dynamic Case Management
Collaboration Tools
"Virtual" data and business rules
Extended Relationship Management
Information, Rule and Relationship Repository
Integration Layer
Application
Application
Application
Application

Fig. 3: Intelligent architecture diagram

Centralized Workflow

An important feature of this platform is a centralized workflow that can span processes, and present a single consolidated action list to the user. This is not easy to achieve. It requires deep integration with the workflow engine of each application; many software products don't expose components of their workflow as web services. Once an item is worked within the action list, it must be removed from the list, and synced with the action list in the target application.

Breaking down silos

A big issue in organizations is the silo'd structure, whereby people are only concerned with their own narrow area of focus. This emerged from the old hierarchical model of organizational management, and has persisted despite the emergence of interoperability tools based on open and accepted standards. Part of the reason it has persisted is that integration between software agents is contrary to the marketing goals of software vendors—many of whom seek to sell the widest breadth of functionality to alleviate the need to integrate. Another reason is the mindset fostered by the single hierarchy management structure—if you report to only one person it stands to reason that the priorities of that one person would take precedence over those of other reporting structures, or even the overall goals of the organization.

This platform is a tool to break down the silos by recognizing the connections between people in the organization across contexts. Processes concerned with management of tasks that cross boundaries can now be recognized and built into reward and feedback mechanisms.

Ownership

It is difficult to assign ownership of a platform whose goal is to break down silos, as assigning ownership is by nature a siloing process. There are many who suggest that this platform should be owned by IT, since (as stated earlier) it complements the changing role of IT. However I challenge IT to be forward thinking enough to want to deploy a platform that is configurable by business users, rather than IT. Just as responsibility should be distributed and shared, ownership of this platform should also be distributed. If there is to be any single ownership, it should be in the "C" suite, as this platform is instrumental in helping companies achieve their strategic objectives.

The Future

In 1965, Intel co-founder Gordon Moore observed that the number of transistors on integrated circuits double approximately every two years.[6] This has held remarkably consistently or increased since that time, and can be expected to continue. As a result, smart phones in our pockets have more processing power than supercomputers of yore. In the not-too-distant future, computers will be wearable. They will be on a shirt button or in an ear insert. Input devices will become more intuitive. Speech recognition has made great strides in recent decades and we can assume that this technology will continue to improve, obviating the need for a keyboard. Eventually, speech recognition will be semantic (rather than syntactic), as it will understand the meaning of what we say. Berners-Lee envisioned a Semantic Web, which the W3C defines as "…a common framework that allows data to be shared and reused across application, enterprise, and community boundaries."[7] This can result in machines that search the web to understand each concept that it encounters.

Futurist and inventor Ray Kurzweil predicts that the confluence of advances in processing speed and non-invasive brain scanning technology will result in "The Singularity"[8] whereby we can reverse-engineer the neural state of the brain on a typical computer within 20-30 years (maybe sooner if President Obama's brain mapping initiative[9] comes to pass). Between now and then, processing power will increase in a continuum, resulting in smaller and more intelligent devices, thus more value placed on uniquely human attributes. MIT's Frank Levy and Harvard's Richard Murnane argue that the automation of business processes has heightened the value of two categories of human skills: "expert thinking—solving new problems for which there are no routine solutions; and complex communication—persuading, explaining, and in other ways conveying a particular interpretation of information."[10]

In 1950, Alan Turing devised the "Turing Test"[11]—whether an observer could distinguish between a conversation with a machine or a real human. To date, no machine has passed the Turing test. How much longer will it be until one does?

The platform proposed here represents a bridge to that future—a place to integrate all of the automated agents at your disposal and replace them as new, more sophisticated technology becomes available.

CONCLUSION—YOUR STRATEGY

Bringing this discussion back to Earth, I began this paper with a discussion about strategy, and that's where it will conclude. What is your strategy? Is it operational in nature—create efficiencies and effectiveness; maximize profits and curtail waste? Is it broader—provide a valuable service for the customer; be a positive force in the universe? In either case, one needs to manage the dynamic and rapidly changing paradigms. This requires an open-ended platform that is ultimately adaptive to changes in business and technological needs.

Epilogue

While out walking her dog Tasha, Jen received a call from Molly, her digital assistant. She touched the top button on her shirt to receive it. "Jen, James called and wants to meet with you at three. Apparently there's a glitch with the Kiplagat deal that he wants to talk through with you." Sometimes Jen forgets that Molly's not a real person. The personality designers did a good job with her.

"You're booked will Gil then to discuss staffing, but that's lower priority, so I can bump Gil to Friday at 2. I've accessed the documents concerning the hold-up with Kiplagat, and opened a sub-case. I can stream them to your retinal projector when you're ready. If the issue's resolved by tomorrow, we can still get sign-off on the deal on Thursday as planned, otherwise I've identified a couple of other windows next week." Jen rattled off a few instructions and ended the call.

Jen can't imagine what life would be like without Molly. It was just a few years ago that people would stagger around like zombies, thumbing at their "smart" phones, eyes downcast on the clumsy devices. It seems as though Molly has a deeper understanding of her motivations than her coarse brain scanning technology can achieve. And she's always agreeable (unlike real *people)! Tasha did her business and they headed home. Now that all of the administrative details of her job are just taken care of, and everything she needs for the knowledge work is at her fingertips, she can live a balanced life and still produce more than enough value to justify her high income. Back home to her sometimes disagreeable, but definitely human, husband and daughter.*

REFERENCES:

[1]: Carr, Nicholas, 'Does IT Matter? Information Technology and the Corrosion of Competitive Advantage', 2003, Harvard Business Review

[2]: Smith, Howard and Fingar, Peter, 'IT Doesn't Matter-Business Processes Do: A Critical Analysis of Nicholas Carr's I.T. Article in the Harvard Business Review', 2003, Meghan-Kiffer Press

[3]: Berners-Lee, Tim, 'Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web', 2000, HarperBusiness

[4]: Harmon, Paul, Keynote and Greenwood, Dominic, 'Goals in the Process Continuum: from BPM to ACM and Beyond', March 20th-21st, 2013, bpmNEXT Conference, Pacific Grove, California

[5]: Altman, Roy, 'People Relationship Management - Completing the BPM Value Proposition', '2009 BPM and Workflow Handbook', and Altman, Roy & Altman, Kenneth, 'Dynamic Clinical Pathways—Adaptive Case Management for Medical Professionals', 2011, 'Social BPM', Future Strategies Inc., FL.

[6]: Moore, Gordon E., 'Cramming More Components Into Integrated Circuits', 1965, Electronics Magazine

[7]: 'W3C Semantic Web Activity', World Wide Web Consortium (W3C), 2011

[8]: $100 million brain mapping initiative proposed by President Barack Obama, April 2, 2013,

http://www.whitehouse.gov/the-press-office/2013/04/02/fact-sheet-brain-initiative

[9]Kurzweil, Ray, 'The Singularity Is Near: When Humans Transcend Biology', 2006, Penguin Books

[10]: Levy, Frank and Murnane, Richard, 'The New Division of Labor: How Computers Are Creating the Next Job Market', 2004, Princeton University Press

[11]: Turing, Alan, 'Computing Machinery and Intelligence', 1950