

# Office Information Systems: A Retrospective and a Call to Arms

**Chandra S. Amaravadi**

Department of Computer Science, Western Illinois University, Macomb, USA  
Email: [c-amaravadi@wiu.edu](mailto:c-amaravadi@wiu.edu)

Received 29 May 2014; revised 20 June 2014; accepted 17 July 2014

Copyright © 2014 by author and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

**Office Information Systems is a field of computer science concerned with studying offices and developing technologies to support office workers. The field initiated in 1982 suffered a gradual demise by 2004. However, many of the themes and issues that motivated the discipline continue to exist today. Office workers continue to struggle with routine tasks that do not have adequate IT support and the problem is worse for mobile employees. In this paper we provide a retrospective on OIS technologies with the hope of stimulating interest in the discipline and addressing the needs of office workers worldwide. The evolution of office technologies and a detailed discussion of past research are provided. The discussion shows that technology support for office workers is still inadequate. The paper suggests a technology vision and architecture to motivate researchers and identifies challenges for future research.**

## Keywords

**OIS, Office Information Systems, Office Systems, Office Automation, Workflow, Evolution of Office Systems, Evolution of OIS, Office Procedures, Office Processes, Office Technologies**

---

## 1. Introduction

Office Information Systems is primarily concerned with technologies to support the work of office workers. It was formerly an academic discipline that involved the study of offices and its application to the development of office systems [1]. The ACM Special Interest Group on Office Automation (ACM SIGOA) was formed in 1979 and the first ACM SIGOA conference was held in 1982<sup>1</sup> (see **Table 1**). In 1983, ACM Transactions on Office Information Systems (ACM TOIS) commenced its maiden issue, acknowledging that it was an interdisciplinary field defined by the topics researched:

*While it may be difficult to describe the area succinctly, there would be a large degree of consensus among*

---

<sup>1</sup>Personal communication with Ms. Irene Frawley, ACM SIG Program Co-ordinator, June 2014.

**Table 1.** Anatomy of a dissolution: A time line.

Anatomy of a dissolution	Major events in the OIS field		
	Date	Event	Comments
	March 1980	First SIGOA newsletter published	Charter: “all aspects of the application of computing techniques to office activities”
	June 1982	First ACM SIGOA conference	To bring together researchers interested in office automation
	January 1983	ACM TOIS starts its inaugural issue	To research office activity and office technology
	June 1986	SIGOA changes its name to SIGOIS	Multiple reasons including attractiveness, goal not being automation, distinguishing from vendor conferences
	October 1988	ACM TOIS last issue	Name change to ACM Transactions on Information Systems in 1989
	December 1996	ACM TOIS last issue	Name change to ACM Transactions on Information Systems in 1989
	November 2004	SIGGROUP dissolved	Due to natural causes

workers in the area as to valid and appropriate topics: models of office information and communication flow, integrated communication systems, acceptance of technology within the workplace, workstation design, innovation in human interfaces, integration of distributed information sources, expert office systems, improved measures of productivity ([2]: p. 1).

A technical focus was also acknowledged:

*Research Contributions will present new and relevant research on topics related to office automation, information, and communication. The papers provide new insights and point to new directions for the development of systems ([2]: p. 2).*

The research that was published ran the gamut of office activity studies, case studies, various office models, office system prototypes, user interface designs etc. Apart from ACM outlets, “Database Journal” and “Journal of Management Information Systems” (JMIS) and “IEEE transactions on Software Engineering” also occasionally published papers in OIS (see for example [3]). In October 1988, ACM Transactions on Office Information Systems changed its name to ACM Transactions on Information Systems. This event had the effect of eroding the identity of the field and paved the way for its gradual demise. The reason given for the name change was a growing feeling among the SIGOIS members that “office” was an outmoded concept and led to the dropping of the term from the title. As explained by [4], “The term ‘office’ is now out of date. People work not only in offices, but at home, on planes, and in the field...” SIGOIS was subsequently renamed SIGGROUP (ibid) in 1996, with the rationale that more of office work was being performed in groups. SIGGROUP was dissolved in 2004 [5] presumably as a result of lack of member interest.

At the present time 27 million employees or 35% of the labor force in the US still work in offices [6]. In spite of the considerable technological support that contemporary office workers enjoy, the state of IT support for office tasks is still very primitive. For instance, consider the job of a training co-ordinator (TC) at a large software consulting company we will call BSS<sup>2</sup>. The TC identifies training needs from the company’s project managers (PM) and develops a monthly schedule of training sessions. He/she then identifies “faculty”<sup>3</sup> to conduct these sessions. The training schedule is placed on the web and interested employees fill a small form online. The TC collects the information from the web and prepares a list of “trainees”. He/she compiles a mailing list and sends a reminder to all trainees, a few days before the start of the session. When the session starts, the TC takes attendance and feedback (if any) and submits them to the Human Resources department. Faculty evaluations are also collected and submitted to management. Although this is seemingly mundane activity, the TC carries it out manually, via a “software bureaucracy”, *i.e.* through an endless number of “point”, “click”, “type”, “copy” and “paste” operations. In his/her present environment, the TC cannot automatically send email to the PMs or compile a trainee list or automatically send reminders (unless a specific web application is built for it). This is a

<sup>2</sup>Based on personal observation while a consultant with BSS. The name is disguised.

<sup>3</sup>Faculty could be external or internal, such as a senior programmer.

classic scenario that is repeated in countless organizations worldwide for reasons that are well known [7]:

- 1) No single software or software combination satisfies all business needs.
- 2) If a software package were available, it would either be difficult to customize or lack the customization features.
- 3) Customized in-house applications as well as business software lack access to functionality of other business software within the same operating environment. Tasks performed by office workers are interrupted at software boundaries.
- 4) Companies lack the resources to develop *ad hoc* IT solutions for the myriad office tasks that arise in their environment.
- 5) Tasks change over a period of time and therefore cannot be pre-coded such as by workflow systems. Often such systems are developed for a specific application e.g. for processing job applicants and cannot be adapted for other applications.

Current desktop software cannot take actions on behalf of the user except in a rigid pre-wired sequence using macros. Users ought to be able to give a command such as “compile a list of employees interested in AI and inform them that there is a seminar on Tuesday”. In conventional systems, this simple task requires a number of actions on the part of the user, which are better performed automatically. Most software impose a bureaucracy in terms of the number of keyboard/mouse actions that need to be taken for even simple tasks.

For employees who do not work in offices, *i.e.* mobile employees the problems are even more compounded because of device limitations and connectivity issues [8]. One mobile employee has resorted to carrying all his (her) office files in a laptop so as to have ready access to information.

It is the thesis of this paper that there are still many unresolved issues in the OIS field that need to be addressed within the auspices of a bounded discipline. The aim of this paper is to provide a context for this discussion and to highlight these issues. The focus of this work is office information systems from a technical (systems) point of view, so we will start the discussion with a historical perspective on OIS technologies.

## 2. Brief History of OIS Technologies

Office automation took its birth in the late seventies against the backdrop of maturing mainframe technologies (see [Table 2](#)). When the IBM PC was introduced, the stage was set for personal software aimed at the office. From [Table 2](#), it is clear that the evolution is marked by the introduction of various hardware and software technologies. Rather than to discuss these individually, it is convenient to discuss them in terms of generations.

The first generation systems were standalone software such as word processing and spreadsheets developed soon after the introduction of the IBM PC [9]. These were single-user generic programs designed for activities such as typing letters/documents and preparing financial statements. These can be regarded as the PC version of mainframe technologies such as PROFS and CEO. Lotus 1-2-3 and Word star exemplify this generation (see [Table 2](#)). The second generation systems were multi-function systems such as word processing, database and spreadsheet, integrated into a single package [9]. We can regard them as integrated extensions of the first generation. Systems such as Symphony and Open Access (see [Table 2](#)) are part of this generation [10]. Systems integrated to a lesser extent, such as through common operating environments also fall into this umbrella. For example, operating environments such as MS-Windows were part of this generation although the concept did not become entrenched into the desktop until the 90's. Workgroup and network-based systems comprise the third generation systems. These systems link together groups of employees working on common projects and enable them to share information. Email, Distributed CASE tools, and similar groupware belong to this generation. The co-ordinator, first introduced in 1985 is a well known example of the third generation [11]. In this system, work group communications were viewed as “commitments”, “obligations”, “propositions” etc. The system was a forerunner for more sophisticated groupware. Given the technological focus hitherto, none of the three generations of office systems had the ability to specify and automate business processes. So they were incapable of supporting process-oriented office activity such as approving loans or handling work orders. They were primarily generic tools utilized in segments of processes and adapted to the tasks by relying on the limited customization capabilities provided. While it was expected that the fourth generation systems would have some intelligence, the nineties have witnessed the birth and explosion of a new category of office software known as workflow systems [12]. Flowmark is one of the well known products in this area [13]. These systems do not have intelligence, although the capability to perform tasks automatically or to act in accordance with the roles

**Table 2.** The evolution of OIS technologies: A time line.

Evolution of OIS technologies	Major technology events in OIS		
	Year	Event	Comments
	1981	“CEO” by Data General	An integrated package offering mail, calendar, word processing [14]
		Introduction of IBM PC	The PC signaled platform change [15]
	1982	IBM Profs introduced	An integrated package for email, calendar and word processing inspired by CEO [15]
	1983	Lotus 1-2-3, WordStar introduced	Lotus 1-2-3 and WordStar are PC-DOS based spreadsheet and word processing software
		DEC All-in-1	All-in-1, a mini-computer based software combined email, group scheduling, calendaring, directory, word processing and filing [16]
		GoldenGate	Goldengate has spreadsheet, database, word processing, graphics and mainframe-to-pc communications for data downloads [10]
	1984	Lotus Symphony	A PC-DOS based integrated suite consisting of spreadsheet, word processing, graphics, communications and database modules with add-ons [17]
		Open Access	Also includes a spreadsheet, word processing, calendar, database and communications [10]
		Framework	Includes word processor, spreadsheet, graphics and a rudimentary database [10]
	1985	Co-ordinator	A groupware package that includes word processing, calendar and communications based on language action perspective [11] [18]
	1988	Token ring LAN	Initial platform for Group Decision Support [15]
		Group Decision Support Systems	Group decision support systems support group activity such as brainstorming, issue identification and analysis and voting [19]
	1991	Lotus Notes	A distributed document database [20]
	1992	Action Workflow	A workflow system based on the co-ordinator [21]
	1994	Internet released	It [22] eventually resulted in a new platform for office technologies
	1995	FlowMark	A workflow system supporting process definition, execution and control [13]
		Netscape navigator released	This technology [23] paved the way for intranets and web-enabled workflow
	2007	Apple iPhone	Paved the way for mobile desktops [24]

and responsibilities of office workers may give a semblance of this ability. The main focus of the fourth generation systems is automation of business processes. By enabling employees to create process models and operationalize them, fourth generations systems are providing much needed support in areas such as loan approval, order processing, purchasing etc. Many of these are web enabled for better accessibility.

Fifth generation systems (we will refer to these as next generation OIS) do not exist leaving open the possibility for conjecture. According to some researchers they are expected to be integrated in function (as with the second generation), capable of process support (as in the fourth generation), and will likely include a primitive level of intelligence [25]-[27]. The systems will likely have the ability to parse and fulfill primitive natural language requests for both actions (*i.e.* processes) as well as for information. For e.g. an employee may request the system to “apply for leave on the 13th”, and the system would carry out the task. These systems would have the ability to learn about and recognize objects in the end user’s universe such as his/her superiors, colleagues, contacts, reports etc. They would also be specifically designed to handle minor problems such as re-scheduling meetings for an employee if he/she goes on leave. Such systems will need to exploit contemporary technologies such as tablets, mobile phones, cloud computing, etc.

### 3. A Survey of Past Research

In this section, we will briefly summarize and highlight some of the research carried out in the OIS field as de-

fined in the earlier sections. As it is not possible to do justice to all the work that has been carried out, the reader is referred to [3] [7] [28] for more detailed discussions of the literature. There are two major streams work in the OIS literature which we will label as “office activity studies” and “modelling research”.

### 3.1. Office Activity Studies

There have been a number of studies of offices, both qualitative as well as quantitative. The quantitative studies attempted to identify the proportions of time office workers spent on various activities such as typing, filling forms, reporting etc. [29]-[34]. The qualitative studies reported on problem solving processes in organizations [35] [36]. A majority of the studies date back to the seventies and eighties. The usefulness of the studies is limited by their purpose as well as the underlying paradigms used to collect the data. The early studies were intended to identify the scope for office information systems and were therefore focused on the activity paradigm. This meant identifying the proportion of time spent in each activity. Even within this paradigm, there was a tremendous variation in the terminology, owing obviously to the infancy of the field. For instance, one study classified office activity into “advising”, “deciding”, “approving”, “arranging”, “scheduling”, etc. while another classified it into “writing”, “proof reading”, “calculating”, “mail handling”, etc. The percentages of time spent in each activity are themselves not very useful and this is compounded by the differences in terminologies so that meta-data analysis cannot be carried out. Since the office has undergone radical changes, these studies will be of limited utility at present and new studies have to be undertaken.

The qualitative studies focused on the co-operative nature of office life required especially to interpret policies and to resolve problems [35] [36]. Gerson and Star [35] reported on the due process (“articulation”) that accompanies the pricing and classification of medical services while Suchman [36] reported on the due process that accompanies the troubleshooting of purchase orders in an accounting office. Both these studies highlight the richness that seems to characterize office work. Classification of medical procedures or troubleshooting of purchase orders requires “articulation”, the set of activities that are required to perform a task. These include discussion, negotiation and information exchanges with various entities within and beyond the organization. Within the realm of OIS literature, there are few studies of this type pointing to a research lacunae.

### 3.2. Modelling Research

The majority of the OIS literature is concerned with office models, reflecting their important role in architecting office systems. The traditional modeling literature dates back to the eighties; in the nineties, this was diffused into workflow models (process models) and then there has been a gap in the literature. The literature is stratified into a number of areas including: 1) “forms”, 2) “documents”, 3) “agents”, 4) “procedures” or “processes”, and 5) “problem solving”. The literature is characterized by specialized approaches within these areas. These are reviewed in the following paragraphs.

#### 3.2.1. Forms

In the forms perspective, database models of forms and specifications of form operations have been developed [37]-[40]. Forms models are generally adaptations of basic hierarchical or relational data structures which consist of groups of attributes that are mapped to base tables. The external presentations of forms correspond to database views and include fields and labels as well as other information such as instructions, page numbers etc. Form operations include form filling, filing, update, copying, retrieval and transmission. The interface aspects of these operations and integrity problems that can arise have been addressed [37]. The representation of forms is still an area needing research. The conventional approach of storing form data using normalized base tables is awkward because of the interdependence among attributes and the need to model operations on them. The problem becomes more complicated for attributes which require simultaneous visibility of other form attributes. In tax forms for instance, the tax rate will be based on filing status and gross wages which are filled earlier in the form. Frequently, attributes of a form such as capital gains in a tax form are looked up from other forms or tables. To build intelligence into forms applications therefore requires a conceptual representation that is closer to its actual presentation than a secondary representation where data is partitioned into a number of normalized tables. A conceptual level view [41] having all the form’s data types such as date-ranges, income classification, data aggregates etc. is more appropriate for advanced forms applications (see [42] p. 54 for exemplary data types). Such data types will suitably be identified from a thorough research into forms structures and data filling

operations. An academic study and documentation of the business logic of form operations is also warranted.

### 3.2.2. Documents

A number of data and document models have been introduced into the OIS literature which have been inspired by semantic data models [43]-[47]. These models address three main aspects of data modelling: classification, object definition/relationships and activities. As in object-oriented models, data objects have been classified into types-*structured* objects such as forms, memos, financial worksheets etc and *unstructured* objects which have free form text and graphics. Some models such as Minstrel-ODM [43] and TEMPORA [45] allow for non-exclusivity in the classifications, resulting in a network structure of object types (rather than a hierarchy). The definitions of object types make use of such well known mechanisms as abstraction, reference, object composition, object instances, derivation, time-stamping etc. In TEMPORA for instance, objects can be simple, composite or derived. Composite objects are those that consist of other objects; for instance, a document could consist of “author”, “abstract”, “table” etc. each of which are individual objects. The details of composite objects are not shown in the main schema but in the object’s sub-schema. Derived objects are specified by constraining simple or complex objects with conditions. For instance, a proposal can be viewed as a document that is “submitted for evaluation”. Objects can have a variety of relationships with other objects such as “part-of”, dependent, independent, exclusive etc. Dependent objects for instance, are objects whose existence depends on other objects. Thus a “car engine” is a dependent object of “car” [45]. The activities performed with objects usually include storage, presentation, retrieval, deletion and update [48]. There has been a tendency in the research to define these activities in terms of their relational equivalents which causes problems similar to those in forms *viz.* the inability to define higher level activities. The representation must be close to the native form of the objects or operations on them become awkward. A second major criticism is that the implementation perspective has been missing from many of these approaches. This is perhaps due to the fact that maintaining the various types of objects and their relationships while ensuring the fulfillment of integrity constraints is an extremely complex problem. Additionally, many of the operations, such as querying will depend on the object type. In recent years there has been a push towards representing digital documents using markup languages. The markup languages are domain dependent and have been developed for certain domains such as bibliographic databases [49]. But generalized semantic models fulfilling a variety of data needs in practical office situations continue to be a research challenge. Models of document operations which go beyond simple retrieval such as, extracting parts from different documents are also potentially of interest to next generation systems.

### 3.2.3. Agents

Procedures are executed by human agents within the context of the goals and policies of the organization. Agent’s roles and responsibilities have been modeled using sets, predicates and objects [50]-[52]. An interesting approach is to model the different roles of an agent (such as “manager”, “team member”, “sub-ordinate”) with a set of properties describing the agent, the states which the agent could have (such as “assigned”, “re-assigned”, “suspended”, “terminated”), the set of messages that the agent could receive that would move the agent to the different states (for e.g. “assign”, “on leave”) and the rules for transitioning among the states (“if leave is approved suspend-role (project manager)”) [52]. Goals and policies have been modeled with the Actor Language (similar to LISP), with rules, and with logic [51] [53] [54]. Modelling of policies does not appear to present any special problems given the variety of languages available today, however, the manner in which they influence processes is still of research interest. For instance, how do organizational policies influence the hiring of a contractor? Are policies implemented implicitly by the agents or explicitly? At what stage in the process are they an issue? Implementation research could take the approach of identifying and developing policy objects accepting a given situation as input and assessing whether or not a policy is violated. Can a part time employee work for more than 10 hours of overtime in a given week? Agents communicate with other agents and with applications, but research in these areas under the auspices of the OIS field has been comparatively sparse.

### 3.2.4. Processes

In the process perspective we dovetail traditional work on procedures with workflow research. Processes have been modeled using both graphical and declarative techniques. Graphical specifications are usually variations on Petri-Nets, Data flow diagrams, State transition networks or activity networks [7] [55] [58] while declarative techniques specify processes using programming languages [51] [52] [58]. A similar pattern is found in recent

work where researchers have used Petri-nets, variations on state transition networks, program specification techniques, and additionally transaction models, UML, logic and frames [59]-[64]. The criticisms of these approaches are best described by the concepts of domain adequacy and representational adequacy [7]. Domain adequacy refers to the extent to which the models incorporate all elements of a process while representational adequacy refers to the extent to which the model incorporates the control structures necessary to specify the process. Fortunately, both have been defined by the Workflow Management Coalition (WFMC) [42]. The domain dimension includes “activities”, “participants” (*i.e.* agents), “transitions” (*i.e.* pre- and post conditions), “workflow relevant data” (data needed by the application as well as decision data) and “application assignment” (*i.e.* a module which executes the activity). The representational dimension includes basic control constructs such as “in-line block”, “loop”, “split”, and “join”. For full details, readers are referred to WFMC [42]. There is an additional dimension to representational adequacy, a concept best described by the three-schema architecture (TSA) in databases [41] [65]. External models are views of processes, conceptual models are those where there is a 1:1 correspondence between the model and the domain, while internal models are implementation level models and often manifest in the form of software specifications. The perspective that models need to have different levels of abstractions has often been missing in the literature. Evaluating models along the TSA and identifying effective practices is clearly a pressing research task. Additional process issues are discussed subsequently.

The realization that business processes are not as straightforward as they seem [35] led to a number of attempts at dealing with problems and exceptions. These are hypothesized to occur as a result of missing information, missing employees etc. Also known as adaptive workflow, problem solving approaches have attempted to rely on artificial intelligence techniques such as consistency checking, hierarchical planning and alternative goal/agent specification [55] [58] [66]-[68]. This is an agenda that is subjected to the well-known challenges in AI such as problem-specificity, goal specification, dynamic specification of goals, representing the different problem states and specifying the conditions under which alternative goals may be sought. Understanding the process of problem solving as it is carried out in offices is a more achievable research goal requiring both quantitative and qualitative evidence on the nature, extent and types of problem solving.

## 4. New Technologies for OIS

The previous section identified a number of research issues including activity studies of offices, form operations, document models based on complex data types, policy agents and nature of problem solving as areas that require further study. However, these are issues within individual areas. There have been few integrated approaches [7]. The major problem is that current systems are incapable of *functional support* for office workers or the capability to perform *ad hoc* tasks across software boundaries. The solution is to take advantage of contemporary technologies and re-architect office systems using a service-oriented model and then to add intelligence so that office systems will function like assistants to the employee. In the following paragraphs technologies needed to enable this vision are discussed.

### 4.1. Mobile Computing

Mobile phones have evolved from simple communication devices to very sophisticated multi-function devices that are already part of the organizational computing landscape [69]. As many as 90% of smart phone users access organizational resources through their devices. However, mobile phones have been integrated much more rapidly into their users’ daily lives than into their work lives. They are still awkward to use for many office tasks such as developing spreadsheets or preparing formal reports. This is because speech is the natural interface for mobile phones rather than the keyboard-monitor interface common for desktops. It is for this reason that voice interfaces for application functionality are of great importance to the future of mobile computing.

### 4.2. Pervasive Computing

Pervasive or Ubiquitous computing is the creation of environments saturated with computing and communication capabilities that are seamlessly and unobtrusively integrated with human users [70]. They make use of sensors, mobile and cloud resources to help users react dynamically to situations. Pervasive computers can help open doors automatically, report on parking situations, detect when a person is busy or help a cell phone user manage his/her health [71]. The most significant problem in pervasive computing is context recognition and many strides have been made in this direction. Areas such as user’s physical location, activity classification,

scene classification and object recognition have been addressed [72]. Context recognition technology will be useful in OIS for understanding a user's request and this is expected to be an easier task since the scope is limited to office work.

### 4.3. Speech Recognition

Speech Recognition is called SR or ASR (Automatic Speech Recognition). The basic problem in SR is to map a speech signal into a set of words in the spoken language. The first step is to identify “phonemes” or pronunciation sounds that are combined to form words. Phonemes are identified through one of two techniques, Hidden Markovian Models or through neural networks. The former approach relies on sampling the speech signal for small time intervals, transforming the signal and using state transition networks to identify phonemes. The latter approach uses multi-layered nets for acoustic modeling of speech signals and requires very large volumes of training data [73]. The next step is word recognition and for this the system needs a vocabulary for matching. Despite decades of research, SR is still a very complex, computationally intensive task that is confounded by vocabulary, context and background noise, all of which will exist in mobile phones.

### 4.4. Service-Oriented Architecture

Service-oriented architecture (SOA) is a mature concept in the software engineering field. It is simply the technique of re-organizing software applications into a set of interacting services that could be made ubiquitously available [74]. A service could be a simple request to verify a user's login name or it could be an entire business process such as an order for parts. In a distributed setting, a service could be accessed and executed on a number of different platforms making it an ideal technology for OIS applications. To find and execute services in a distributed setting requires protocols between providers and service clients. SOAP is one such popular protocol in use on the internet.

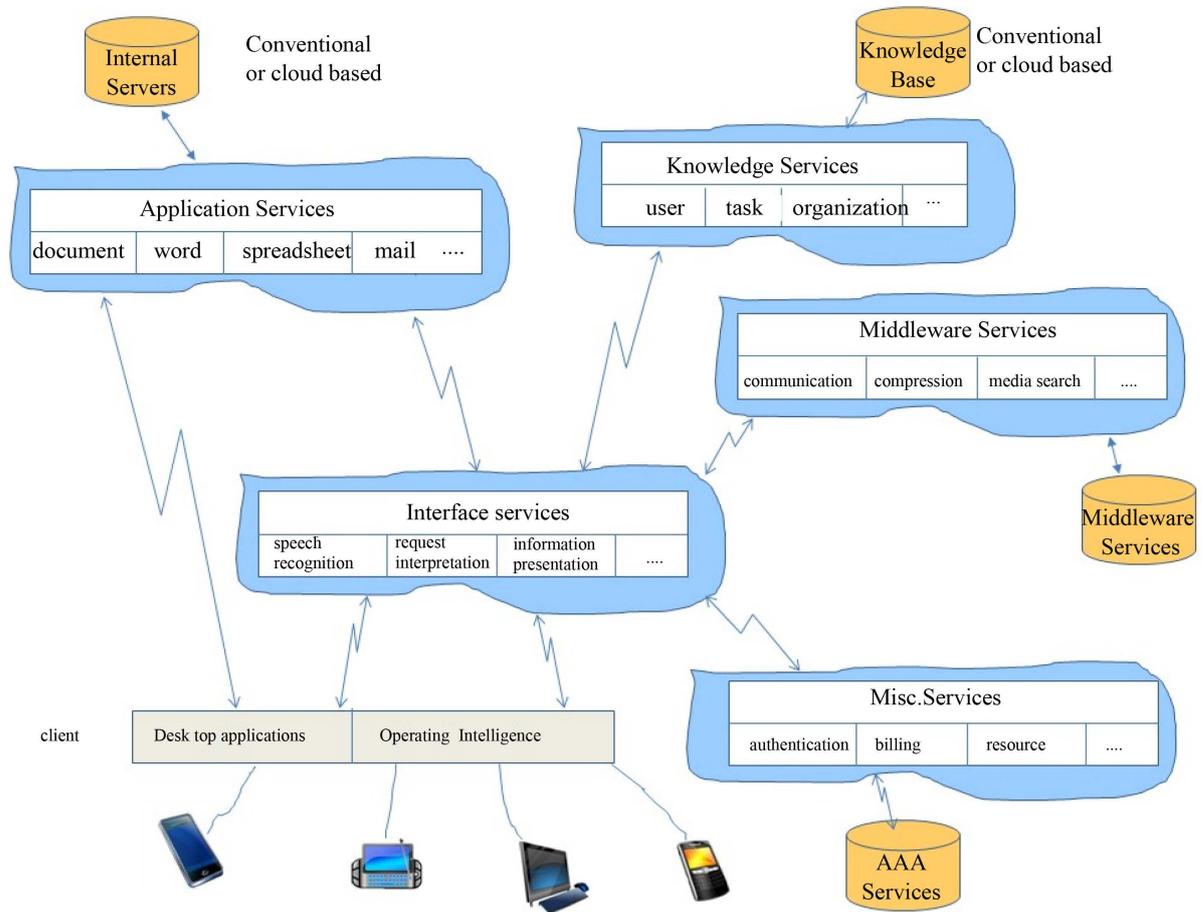
### 4.5. Cloud Computing

According to [75], cloud computing is a model enabling convenient, on-demand network access to a shared pool of configurable computing resources. There are three types of cloud services—infrastructure, platform and application [ibid]. Infrastructure is a very basic type of cloud service and is concerned mostly with storage and network services. Platform services provide a development environment for running and testing applications. The last is also known as SAAS (software as a service) and is concerned with providing applications on demand to users. For OIS, a combination of these type of services could be utilized. For example office software could be cloud-based while at the same time some infrastructure services such as video storage and or information extraction could also be provisioned from the cloud.

## 5. A Renewed Vision for OIS

In this section a new vision for OIS is proposed, based on extending ideas previously proposed [25]-[27] using contemporary technologies. The next generation OIS will play an integral part in the employee's daily work and will act like their assistant. With the system, office workers will be able to enter voice commands to carry out actions that could range from information requests to workflows—“Whom should I contact to draft a contract?” “Who in engineering is an expert in ball bearings?”, “Apply for leave from July 4<sup>th</sup>-20<sup>th</sup>”. These requests are understood through the use of user profiles and task knowledge stored in the system. Employees will continue to have access to traditional applications such as Word Processing, SpreadSheet and email. They will also be able to issue such requests from any of these applications, using a natural language/voice interface embedded in all office software. This will provide access to the features of an application from another application, such as sending mail while working on a presentation. Mobile employees will be able to access all office functionality subject to limitations of their devices. They will also take advantage of cloud resources to overcome such limitations. An architecture that can potentially fulfill these requirements is shown in **Figure 1**.

The diagram shows that the desktop continues to exist as before and direct access to applications is allowed as before. There is a new component called the Operating Intelligence (OI) that will be ubiquitously accessible. The OI can be directly invoked from the desktop or from within an application. Users can interact with the OI through voice or natural language commands. Five types of requests may be made:



**Figure 1.** A contemporary vision for OIS.

- 1) Request for information in documents—Users can find out information stored on their machines such as a file having the title “New Jersey Project”.
- 2) Request for storing miscellaneous information/knowledge such as a meeting at a particular time or the name of a contact or that a particular vendor offers discounts till March [76].
- 3) Queries based on system knowledge—This is concerned with the knowledge that the OIS has regarding the organization and its employees. “Do we have a sales office in Delaware?”, “How many employees work there?” [ibid].
- 4) Request for applications—This request may be for conventional applications such as MS Word, Powerpoint etc. or for specific applications such as video editing software. The latter type requests will ideally be provisioned from the cloud.
- 5) Request for action—This is concerned with taking simple actions in the software on behalf of the user or more complex actions. An example of the former is “highlight third paragraph in the LG group proposal”. An example of the latter is “highlight third paragraph in the LG group proposal and send it to the author to revise”.

Execution of these requests requires the architecture components shown in the diagram (please see **Figure 1** again).

*Interface services* include speech recognition, interpretation as well as information presentation. Information presentation can refer to multi-modal presentation such as voice-to-text, text-to-voice, video-text etc. as well as mapping between desktop to mobile device and vice versa. In addition, it becomes necessary to save the user’s context if they move across devices (*i.e.* distributed interfaces). Being hardware level, these issues are well understood and are being addressed [77]. *Context services* (not shown) require intelligence and are used to identify the user’s location, task, activity etc. as noted previously. Many smart phones have location services built-in

with add-ons for activity monitoring.

*Application Services* refers to componentized desktop applications such as “send mail”, “get mail”, “update file”, “change heading style in XYZ powerpoint”. Two issues that arise here are the interface specifications and dynamic configuration of services to answer a request. Also one assumes some sort of cloud-based utility infrastructure already present, to provide applications on demand. Billing and authentication for such services is grouped under *miscellaneous services* and is also a relevant research topic [78].

*Knowledge Services* provide knowledge/information to any user or to any application. As discussed above, a knowledge query could be simple such as “what files did I create last month?” or more sophisticated such as “who is the expert in Oracle Financials?” Knowledge services require knowledge of the user, task, environment, system etc. There is some overlap with context identification services. It should be noted that knowledge services encompasses the functionality provided by knowledge management [76]. This requires analysis, encoding and classification (ontology) of organizational knowledge [ibid]. There have been some proposals for handling such knowledge in the literature [25] [76] [77]

*Middleware Services* are required for mobile clients to overcome resource limitations inherent in mobile devices. Examples of such services include communications, compression, security, media search etc. A wireless client could request the first 10 seconds of a product advertisement. Use case studies are needed to identify middleware requirements. Such services are generic and can be utilized by any type of mobile employee. Apart from service identification, the logistics for delivering such services and performance are issues needing research. For a discussion of these issues please refer to [78].

## 6. Conclusions

The paper presented the evolution of office technologies and has demonstrated that neither these nor past research has brought office technologies closer to the task of supporting office professionals. The fundamental problem appears to be the way in which contemporary desktop technologies are conceptualized. These are monolithic applications with rich functionality within that particular environment. The solution to this has been presented as a vision and architecture to motivate researchers. Task support requires a rich combination of technologies (rather than rich functionality) that are dynamically configured, ubiquitously available and responsive to voice commands. The solution is to re-conceptualize desktop applications in terms of services that are available from any application. Services are requested and executed through the Operating Intelligence. Service requests can be understood and executed only within the context of the user environment and this requires context knowledge. Thus knowledge services are also required. Finally since mobile environments are associated with particular challenges, mobile services such as downloading large files and presenting individual pages of large documents are also required. Apart from those challenges already pointed out, identifying a software architecture corresponding to the functional architecture that has been presented will be an immediate challenge. The scope of the present paper was such that this issue could not be addressed.

An integrated approach as described has not been previously conceptualized and realized as a prototype in the literature. This is a problem of specialized approaches in the field coupled with the fact that the field as a whole was dying by the early 2000’s. Additionally, the technologies and infrastructure to make the approach feasible did not mature until recently. The approach is technically feasible since the underlying technologies have already been applied.

## References

- [1] Amaravadi, C.S. (1989) Towards a Conceptual Model of the Office: An Integrating Approach. Ph.D. Dissertation, The University of Arizona, Tucson.
- [2] Limb, J. (1983) Editor’s Introduction. *ACM Transactions on Office Information Systems*, **1**, 1-2. <http://dx.doi.org/10.1145/357423.357424>
- [3] Desai, S. (1991) Unification of Underlying Concepts in Different Office Models. *DataBase*, **22**, 38-45.
- [4] Swenson, K. (1996) SIGGROUP Name Change. *SIGOIS Bulletin*, **17**, 3.
- [5] Walker, R. (2004) Notice of SIGGROUP Dissolution. <http://www.acm.org/sigs/siggroup/siggroupinfo.html>
- [6] Anonymous (2013) Office Buildings. [http://www.eia.gov/emeu/consumptionbriefs/cbcs/pbawebbsite/office/office\\_howmanyempl.htm](http://www.eia.gov/emeu/consumptionbriefs/cbcs/pbawebbsite/office/office_howmanyempl.htm)

- [7] Amaravadi, C.S., George, J.F., Sheng, O.R.L and Nunamaker, J.F. (1995) The Adequacy of Office Models. In: *Advances in Computers*, **40**, Academic Press, New York, 182-247.
- [8] Makinen, S. (2012) Mobile Work and Its Challenges to Personal and Collective Information Management. *Information Research*, **17**, 522 .
- [9] Tschritzis, D. (1986) Office Automation Tools. In Jarke, M., Ed., *Managers, Micros and Mainframes*, John Wiley and Sons, New York, 11-20.
- [10] Amaravadi, C.S. (1985) An Examination of Integration Alternatives in Micro-Based Workstations. Technical Report, University of Arizona, Tucson.
- [11] Winograd, T. (1987) A Language/Action Perspective on the Design of Cooperative Work. *Human-Computer Interaction*, **3**, 3-30. [http://dx.doi.org/10.1207/s15327051hci0301\\_2](http://dx.doi.org/10.1207/s15327051hci0301_2)
- [12] Georgakopoulos, D., Hornick, M. and Sheth, A. (1995) An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and parallel Databases*, **3**, 119-153. <http://dx.doi.org/10.1007/BF01277643>
- [13] IBM (1997) Sample Applications for Flowmark. Image and Workflow Library, IBM Redbooks, IBM, Armonk.
- [14] Wikipedia (2013) CEO by Data General.
- [15] IBM (1998) IBM Internet Support Group. [askibm@vnet.ibm.com](mailto:askibm@vnet.ibm.com)
- [16] Email Museum (2013) Quick History of All-in-1. <http://email-museum.com/2011/09/26/quick-history-of-all-in-1/>
- [17] Wikipedia (2013) Cullinet.
- [18] Action Technologies (1998) Customer Relations. [Customerrelations@actiontech.com](mailto:Customerrelations@actiontech.com)
- [19] Dennis, A.R., George, J.F., Jessup, L.M., Nunamaker Jr., J.F. and Vogel, D.R. (1988) Information Technology to Support Electronic Meetings. *MIS Quarterly*, **12**, 591-624. <http://dx.doi.org/10.2307/249135>.
- [20] Parsons, T. (1997) Lotus Development. Personal Communication. [Parsons/PHL/Lotus@lotus.com](mailto:Parsons/PHL/Lotus@lotus.com)
- [21] Medina-Mora, R., Winograd, T., Flores, R. and Flores, F. (1992) The Action Workflow Approach to Workflow Management Technology. *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, Toronto 281-288. <http://dx.doi.org/10.1145/143457.143530>
- [22] Leiner, B., Cerf, V.G., Clark, D., Kahn, R., Kleinrock, L., Lynch, D., Postel, J., Roberts, L. and Wolff, S. (2013) Brief History of the Internet. The Internet Society. <http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet>
- [23] Heater, B. (2011) Browsers: A Brief History. <http://www.pcmag.com/slideshow/story/262125/browsers-a-brief-history/3>
- [24] Macworld (2007) Hello iPhone.
- [25] Amaravadi, C., Sheng, O.R., George, J.F. and Nunamaker, J.F. (1992) AEI: A Knowledge-Based Approach to Integrated Office Systems. *Journal of Management Information Systems*, **9**, 133-163.
- [26] Ellis, C.A. and Naffah, N. (1987) Design of Office Information Systems. Springer Verlag, New York.
- [27] Faidt, K. and Karagiannis, D. (1990) Knowledge-Based Applications in Office Information Systems: An Integration Approach. In: *Database and Expert Systems Applications Conference (DEXA)*, Vienna, 29-31 August 1990, 334-339. [http://dx.doi.org/10.1007/978-3-7091-7553-8\\_54](http://dx.doi.org/10.1007/978-3-7091-7553-8_54)
- [28] Mahling, D.E., Craven, N. and Croft, W.B. (1995) From Office Automation to Intelligent Workflow Systems. *IEEE Expert*, **10**, 41-47. <http://dx.doi.org/10.1109/64.393142>
- [29] Christie, B. (1985) Human Factors of the User System Interface. North Holland, New York, 23-70.
- [30] Engel, G.H., Groppuso, J., Lowenstein, R.A. and Traub, W.G. (1979) An Office Communication System. *IBM Systems Journal*, **18**, 403-431.
- [31] McLeod, R. and Jones, J.W. (1987) A Framework for Office Automation. *MIS Quarterly*, **11**, 87-104. <http://dx.doi.org/10.2307/248830>
- [32] Morgenbrod, H.G. and Schwaertzel, H.G. (1980) The Degree of Office Automation and Its Impact on Office Procedures and Employment. *Computer Networks*, **I-II**, 3-11.
- [33] Poppel, H.L. (1982) Who Needs the Office of the Future? *Harvard Business Review*, **60**, 146-155.
- [34] Thachenkary, C.S. and Conrath, D.W. (1982) The Office Activities of Two Organizations. In: Naffah, N., Ed., *Office Information Systems*, North Holland, New York, 452-469.
- [35] Gerson, E.M. and Star, S.L. (1986) Analyzing Due Process in the Workplace. *3rd ACM SIGOIS Conference*, Providence, 6-8 October 1986, 70-78.

- [36] Suchman, L.A. (1983) Office Procedure as Practical Action: Models of Work and System Design. *ACM Transactions on Office Information Systems*, **1**, 320-328.
- [37] Bernal, M. (1982) Interface Concepts for Electronic Forms Design and Manipulation. In: Naffah, N., Ed., *Office Information Systems*, North Holland, New York, 505-518.
- [38] Lum, V.Y., Choy, D.M. and Shu, N.C. (1982) OPAS: An Office Procedure Automation System. *IBM Systems Journal*, **21**, 327-350. <http://dx.doi.org/10.1147/sj.213.0327>
- [39] Tsichritzis, D. (1982) Form Management. *Communications of ACM*, **25**, 453-478.
- [40] Yao, B.S., Hevner, A.R., Shi, Z. and Luo, D. (1984) Formanager: An Office Forms Management System. *ACM Transactions on Information Systems*, **2**, 235-262. <http://dx.doi.org/10.1145/1206.357413>.
- [41] Hoffer, J., Ramesh, V. and Topi, H. (2013) Database Management. 9th Edition, Benjamin/Cummings, New York, 53-55.
- [42] WFMC (1999) Workflow Management Coalition. Interface 1: Process Definition Interchange and Interchange Language, Document # WFMC-TC-1016-P v 1.1. <http://Wfmc.org>
- [43] Harper, D.J., Dunnion, J., Sherwood-Smith, M. and Van Rijsbergen, C.J. (1986) Minstrel-ODM: A Basic Office Data Model. *Information Processing and Management*, **22**, 83-107. [http://dx.doi.org/10.1016/0306-4573\(86\)90117-2](http://dx.doi.org/10.1016/0306-4573(86)90117-2).
- [44] Lamersdorf, W., Schmidt, J.W. and Muller, G. (1986) A Recursive Approach to Office Object Modelling. *Information Processing and Management*, **22**, 109-120. [http://dx.doi.org/10.1016/0306-4573\(86\)90118-4](http://dx.doi.org/10.1016/0306-4573(86)90118-4).
- [45] Loucopoulos, P., Mcbrien, P., Schumacker, F., Theodoulidis, B., Kopanas, V. and Wangler, B. (1991) Integrating Database Technology, Rule-Based Systems and Temporal Reasoning for Effective Information Systems: The TEMPORA Paradigm. *Information Systems Journal*, **1**, 129-140. <http://dx.doi.org/10.1111/j.1365-2575.1991.tb00032.x>.
- [46] Pernici, B., Barbic, F., Fugini, M.G., Maiocchi, R., Rames, J.R. and Rolland, C. (1991) C-TODOS: An Automatic Tool for Office System Conceptual Design. *ACM Transactions on Information Systems*, **7**, 378-419. <http://dx.doi.org/10.1145/76158.76893>
- [47] Wang, J.T.L. and Ng, P.A. (1992) TEXPROS: An Intelligent Document Processing System. *International Journal of Software Engineering and Knowledge Engineering*, **2**, 171-196. <http://dx.doi.org/10.1142/S0218194092000099>.
- [48] Ang, J.S.K. and Conrath, D.W. (1993) The Ithaca Office Object Model: Modeling and Implementation. *ACM SIGMIS Database*, **24**, 5-14.
- [49] Ram, S., Park, J. and Lee, D. (1999) Digital Libraries for the Next Millennium: Challenges and Research Directions. *Information Systems Frontiers*, **1**, 75-94. <http://dx.doi.org/10.1023/A:1010021029890>
- [50] Aiello, L., Nardi, D. and Panti, M. (1984) Modeling the Office Structure: A First Step towards the Office Expert System. *ACM SIGOA Conference*, Toronto, 25-27 June 1984, 25-32. <http://dx.doi.org/10.1145/800023.808330>
- [51] De Jong, P. (1987) UBIK: A System for Conceptual and Organizational Development. *Proceedings of the IFIP*, Toronto, 17-19 August 1987, 19-36.
- [52] Pernici, B. (1990) Objects with Roles. *ACM Conference on Office Information Systems*, Cambridge, 25-27 April 1990, 205-215. <http://dx.doi.org/10.1145/91474.91542>
- [53] Lee, R. (1988) Bureaucracies as Deontic Systems. *ACM Transactions on Information Systems*, **6**, 87-108. <http://dx.doi.org/10.1145/45941.45944>
- [54] Loucopoulos, P. and Katsouli, E. (1992) Modelling Business Rules in an Office Environment. *SIGOIS Bulletin*, **13**, 28-37. <http://dx.doi.org/10.1145/134376.134384>
- [55] Ang, J.S.K. and Hong, J. (1994) AMS Formalism: An Approach to Office Modeling and OIS Development. *DataBase*, **25**, 25-38.
- [56] Barbic, F., Ceri, S., Braachi, G. and Mostacci, P. (1985) Modeling and Integrating Procedures in Office Information Systems Design. *Information Systems*, **10**, 149-168. [http://dx.doi.org/10.1016/0306-4379\(85\)90033-X](http://dx.doi.org/10.1016/0306-4379(85)90033-X)
- [57] Ellis, C.A. and Nutt, G.J. (1980) Office Information Systems and Computer Science. *ACM Computing Surveys*, **12**, 3-36. <http://dx.doi.org/10.1145/356802.356805>
- [58] Kreifelts, T. and Woetzel, G. (1986) Distribution and Error Handling in an Office Procedure System. *Proceedings of the IFIP*, Pisa, 22-24 October 1986, 197-208.
- [59] Alonso, G., Agrawal, D., El Abbadi, A., Kamath, M., Guenthoer, R. and Mohan, C. (1996) Advanced Transaction Models in Workflow Contexts. *Proceedings of the 12th International Conference on Data Engineering*, New Orleans, Louisiana, 26 February-1 March 1996, 574-581.
- [60] van der Aalst, W. (1996) Petri-Net-Based Workflow Management Software. *Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems*, Athens, May 1996, 114-118.

- [61] Basu, A. and Blanning, R.W. (1999) Metagraphs in Workflow Support Systems. *Decision Support Systems*, **25**, 199-208.
- [62] Grundy, J.C., Hosking, J.G. and Mugridge, W.B. (1998) Supporting Large-Scale End User Specification of Workflows, Work Coordination and Tool Integration. *Journal of Organizational and End User Computing*, **10**, 38-48.
- [63] Koubarakis, M. and Plexousakis, D. (2002) A Formal Framework for Business Process Modeling and Design. *Information Systems*, **27**, 299-319. [http://dx.doi.org/10.1016/S0306-4379\(01\)00055-2](http://dx.doi.org/10.1016/S0306-4379(01)00055-2)
- [64] Ngu, A., Duong, T. and Srinivasan, U. (1996) Modeling Workflow Using Tasks and Transactions. *Proceedings of the 7th International Workshop on Database and Expert System Applications, Dexa*, Zurich, September 1996, 451.
- [65] Preuner, G. and Schrefl, M. (2000) A Three-Level Schema Architecture for the Conceptual Design of Web-Based Information Systems: From Web-Data Management to Integrated Web-Data and Web-Process Management. *World Wide Web*, **3**, 125-138. <http://dx.doi.org/10.1023/A:1019281629747>
- [66] Dellen, B., Maurer, F. and Pews, G. (1997) Knowledge-Based Techniques to Increase the Flexibility of Workflow Management. *Data and Knowledge Engineering*, **23**, 269-295.
- [67] Karbe, B., Ramsperger, N.G. and Weiss, P. (1990) Support of Cooperative Work by Electronic Circulation Folders. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Los Angeles, 7-10 October 1990, 109-117. <http://dx.doi.org/10.1145/91474.91496>
- [68] Chung, P.W., Cheung, L., Stader, J., Jarvis, P., Moore, J. and Macintosh, A. (2003) Knowledge-Based Process Management—An Approach to Handling Adaptive Workflow. *Knowledge-Based Systems*, **16**, 149-160. [http://dx.doi.org/10.1016/S0950-7051\(02\)00080-1](http://dx.doi.org/10.1016/S0950-7051(02)00080-1)
- [69] Mangalaraj, G.A. and Amaravadi, C.S. (2008) Connecting to the Next Generation Mobile Desktop. *Proceedings of America's Conference on Information Systems*, Toronto, 14-17 August 2008, 71.
- [70] Satyanarayanan, M. (2001) Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, **8**, 10-17. <http://dx.doi.org/10.1109/98.943998>
- [71] Cook, D.J. and Das, S.K. (2012) Pervasive Computing at Scale: Transforming the State of the Art. *Pervasive and Mobile Computing*, **8**, 22-35. <http://dx.doi.org/10.1016/j.pmcj.2011.10.004>
- [72] Pejovic, V. and Musolesi, M. (2014) Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges. Unpublished paper.
- [73] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B. (2012) Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, **29**, 82-97. <http://dx.doi.org/10.1109/MSP.2012.2205597>
- [74] Papazoglou, M.P. and Georgakopoulos, D. (2003) Service-Oriented Computing: Introduction. *Communications of the ACM*, **46**, 24-28. <http://dx.doi.org/10.1145/944217.944233>
- [75] Pallis, G. (2010) Cloud Computing: The New Frontier of Internet Computing. *IEEE Internet Computing*, **14**, 70-73. <http://dx.doi.org/10.1109/MIC.2010.113>
- [76] Amaravadi, C.S. (2005) Knowledge Management for Administrative Knowledge. *Expert Systems*, **25**, 53-61. <http://dx.doi.org/10.1111/j.1468-0394.2005.00294.x>
- [77] Rosner, D., Grote, B., Hartman, K., Hofling, B. and Guericke, O. (1998) From Natural Language Documents to Sharable Product Knowledge: A Knowledge Engineering Approach. In: Borghoff, U.M. and Pareschi, R., Eds., *Information Technology for Knowledge Management*, Springer Verlag, Berlin, 35-51. [http://dx.doi.org/10.1007/978-3-662-03723-2\\_8](http://dx.doi.org/10.1007/978-3-662-03723-2_8)
- [78] Amaravadi, C.S., Mangalaraj, G., Philip, S. and Balijepally, V. (2009) Envisioning the Next Generation Cellular Client. *Proceedings of Conference-IRM*, Al Ain, 21-23 May 2009, 21-23.

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either [submit@scirp.org](mailto:submit@scirp.org) or [Online Submission Portal](#).

