SOFTWARE DEVELOPMENT PARADIGMS FOR COMMUNITY INFORMATICS – TECHNOLOGICAL ASPECTS

A.B. Patki,
Department of Information Technology, Government of India
India

M.D. Kulkarni
Center for Development of Advanced Computing (CDAC), Pune University Campus, Ganesh Khind
India

and

Dhanvanti D. Patki
SIPNA College of Engineering & Technology
India

ABSTRACT

The trends in Information Technology for Masses are indicative of the fact that with the adequate level of Government support in various countries, the Community Information Centers are growing in numbers. The public expectations from Community Information Systems are on a different track in respect of the utility. The paper discusses the basic requirements of future community informatics in the context of information overload and reducing ‘cognitive load’ of the potential user. Authors examine the software engineering methodology and bring out the development challenges needed in the existing practices of Object Oriented philosophy. A futuristic software model for community informatics is also suggested.

INTRODUCTION

Majority of the benefits of Information Technology (IT) revolution are constrained to reach only to IT professionals and other elite sections of society who are well versed with the working of these systems that usually operate in English language. In a diverse and multilingual society like India, a single language system can never become successful without adequate software support for vernacular computing. The Community Information Centers (CICs) create an infrastructure environment to facilitate Community Informatics oriented activities amongst local communities and help indirectly towards:

(i) Promoting local entrepreneurs development
(ii) Improving and enhancing economic and social participation
(iii) Opening up new avenues for raising the economic status of deprived sections in the community

CIC also provides a unique opportunity by creating virtual Internet communities, permitting older people as well as youngsters, to benefit from the Information & Communication Technology (ICT). Creating community information infrastructure to access and utilize information from a variety of sources including sharing skills of old generation people to open up new avenues for utilizing their experience and knowledge is likely to open up new vistas for old age people as well as socially under privileged section of society. Effects of technology oriented infrastructure like electricity, railways and analogues impact of CIC have been discussed by Patki et.al. (2003). The growing interest world over in community capacity building is another land mark with potential to boost the spread of CIC to overcome digital divide and
also restructure the role of old people in multilingual / multicultural (music, dance, literature, food habits, religious & spiritual practices, flora and fauna, health & medicine etc.) society towards knowledge based future. Such initiatives lead to far reaching effects on the potential economic growth pattern of community through formal and informal social networks and human resource development initiative to build social capital. In contrast to the Government sponsored / funded Community Centers, the privately owned and managed Cyber Cafes are practically based on the business entrepreneur model, resulting into the usage of IT infrastructure for EMAIL, Internet browsing facility for students, web surfing for entertainment, and similar one time activities that are repetitive in nature. However, it does not directly provide any opportunity for Community welfare. Unlike Cyber Cafes, the scope, potentials and management options with CICs to introduce Cooperative / Collaborative Computing are much broader.

The focus in the past was merely on the deployment of existing off-the-shelf hardware/software technology in the form of ICT infrastructure, training people, developing computer applications, propagating the mechanism of change through community infrastructure centers. This approach has resulted in a limited success and deeper insight into the study of such initiatives, indicate that technological developments are required to reap full benefits of ICT and merely creating the ICT infrastructure is not likely to serve the purpose beyond a trade off limit.

INVESTMENTS IN COMMUNITY INFORMATICS VERSUS CORPORATE COMPUTING

In democratic country like India, cooperative sector has its own place in the economic development pattern of community. Profit is not the main motive of CIC when viewed & compared with private sector computing facility set-up. Of course, for CICs, sustainability is of paramount importance and it is out-of-bounds so far as competition with other is concerned. Such Community Information Center also has potential for providing rural citizens with information that they were previously unable to access, and that could benefit their socio-economic development. In this context, we must distinguish usage of Information Technology’s power in private companies, corporate houses, members of association / confederation of industries, who had viewed IT as a resource for productivity enhancements. Nicholas Carr (2003) has indicated that if IT as resource is viewed as infrastructural technologies, the competitive advantage ceases with time. It is argued that pattern of corporate IT spending indicate that greater IT expenditures rarely provide superior financial results. As brought out by Carr, the new rules for IT management for the future emphasize on -

- Spend less
- Follow, don’t lead
- Focus on vulnerabilities, not opportunities

While the concerns expressed regarding investment in IT for corporate computing may not be directly applicable to financial investments & management of CICs, it is pertinent to observe the root causes. Carr has not focused on the primary cause of not recognizing the technological development in other non-electronics fields in the second half of twentieth century & not deploying these in IT forefront properly, to overcome the ‘bleak’ chances of forthcoming investments in near future in IT – especially in Corporate Sector. In order to avoid similar fate for community informatics, it is necessary that technological aspects be examined properly. A case of MISNOS technology in the context of Community Informatics is discussed in the following sections so also suggestions are brought out for Microsoft technologies.

OPERATING SYSTEM SCENARIO AND OPEN SOURCE ISSUES

In the past, Operating System was primarily tailored for individual hardware/CPUs. This has resulted into a chaotic situation that made programs developed on one machine totally incapable of running on another. By and large, although UNIX operating system had arrived earlier than the DOS / MS Windows, most of the Community Center / Cyber Cafe users in India have very little (or practically nil) idea about the existence of UNIX and the Internet growth arrived through UNIX route. UNIX
has a variety of profound and diverse concepts developed and perfected over a period of time. UNIX had to grow through a sequence of three phases in its life cycle:

- A product for engineering & scientific community
- Large corporations & government organizations where the robustness of UNIX established it as the OS of choice for database work
- Internet phase of UNIX cycle

In the last decade of twentieth century, Linux – a UNIX workalike (i.e. Linux responds to standard UNIX commands & runs UNIX programs) began its life as the project of a single man, Linus Torvalds. He designed Linux (similar to Minix) operating system, yet freely available. Around the year 1994, Linux version 1.0 was released. Many volunteers helped create add-on software to make Linux popular. Linux is completely open i.e. you get the source code for the Linux operating system and you can make changes to the operating system. For the most part, Linux and its tools follow various standards, such as POSIX compliance. Linux runs binaries created on other PC versions of UNIX. By and large, for many preliminary computing requirements like, word processing, spreadsheets, and telecommunications – Linux is adequate alternative to other PC operating systems. Since Linux is freely available, it attracts the attention of many computer programmers & software developers who release their software to the computer community. However, a major drawback of Linux is the lack of robust applications for everyday use. The client/server architectures and Personal Computer Systems were neither designed nor planned for Community Informatics. Hence, for supporting multilingual applications for Community Informatics, merely font based multilingual support (for EMAIL, Web browsing, Web content development in local language, Word processing and alike) is of limited usage. This shortcoming comes to limelight when large scope vernacular computing software development support needs to be provided through CICs. Of course, there are no such expectations amongst the users of cyber café infrastructure. The missing link is the lack of “cognitive off loading” software support in the existing Microsoft as well as UNIX/LINUX based software to introduce Collaborative computing using the CICs. This is felt as a major bottleneck by user community to participate effectively through CIC infrastructure.

The preceding paragraphs brought out a scenario that whether it is DOS / MS – Windows or UNIX/Linux, all are appearing as ‘frozen’ operating systems i.e. once configured & installed on the computer hardware, the role of OS is to provide support only & OS itself does not get further evolved over a period of its life cycle on the ‘installed computer hardware’. Any diversification from this ‘set’ idea of operating systems software, appears to be a very strange concept amongst software developers & had attracted severe criticism. While Linux development was in its infancy during 1992-94, the team at Department of Information Technology (DIT) was exploring the possibility of operating software environment that would have an inbuilt capability to support Community Informatics. The issues had to focus on OS potential to

- Evolve itself from learning from user behavior patterns (partially similar to user profile) settings in MS – Windows)
- Evolve itself from learning from usage of various application software packages running under the ‘installed’ operating system configuration
- Incorporating Operating systems Shell features on the lines of Reduced Instruction Set Computer (RISC) hardware
- Providing built ins support at Shell & Kernel level for ‘cognitive off-loading’

Soft computing techniques developed during the second half of twentieth century viz. like Neural networks, Fuzzy Logic, Rough Set methodology, Genetic algorithms etc. were found useful. This had resulted into the evolution of a framework for MISNOS (Multimedia Multilingual Information Services Network Operating Software). The emergence of information banking related services & Community Centers for IT for Masses was reported in the seminal paper on MISNOS Technology by R. Bandyopadhyay
(1996). Community Informatics in a multilingual democratic society like India is a fit case for experimenting new operating software environment to explore and reap its benefits to corporate computing in near future due to globalization of corporate business world over. In this context, it is important to carefully study the observation made by Carr viz. “Let the impatient competitors shoulder the high costs of experimentation, and then sweep them past.

SOFTWARE DEVELOPMENT CONSIDERATIONS FOR CIC INFORMATICS DEVELOPMENT & ROLE OF MISNOS TECHNOLOGY

Software development activities under CIC informatics hitherto, primarily included ‘font’ based web content creation for multilingual society without vernacular computing support. By and large, in the absence of suitable avenues, the CIC inhabitants were tailored for a default discipline earmarked for Internet surfer community. The days of ‘font’ oriented activity towards web content creation for multilingual society are numbered. Such web contents fall under the ‘one time referral’ type of information systems. Similar systems have been analyzed earlier by Sivasubramanian (1996) in the context of Software Industry Data Base. However, it has been seen that CIC inhabitants can use the infrastructure for a variety of other useful purposes leading to information banking type of activities. Some of the group projects for local language oriented cultural activities could be in the areas of poetry, drama writing, paintings etc. There is no support software at the CIC for undertaking these activities by CIC inhabitants. It is important to observe that the CIC inhabitant group activities are absolutely different in nature and kind than the Yahoo User Groups. CIC inhabitants even find it enthusiastic to conduct group activities for events like recitation competitions and prize distribution while celebrating community festivals. Such activities call for software development by the community inhabitants who are not very well conversed with computer programming. Moreover to realize the full potential of the ‘IT Revolution’, system development and implementation should be ideally made so simple that with minimal computer training / knowledge, user should be in a position to accomplish the same. MISNOS Technology founder team has observed this as an essential characteristic feature of the future Community Informatics systems. MISNOS technology had suggested a need and given a technology development direction for enabling ordinary users to develop software for their local needs. In this context, it is pertinent to observe that with the introduction of windows systems on Personal Computers, the load of remembering typical Disk Operating System (DOS) commands was removed. Windows provided a Graphical User Interface (GUI) to common user. Subsequent development in window philosophy further eased out other frequently required operational functions. Thus, the population of PC users was increasing through the deployment of window systems. The latest boom brought out by TCP/IP and in Internet browser technology has further resulted in giving a new face to the computerization approach of mass utility. This resulted in the government computerization taking speedy shape in implementing old type of computerization program in the form of e-governance projects. Common measure of IT penetration was considered as PC density and Indian IT penetration is at 5.8 PCs per 1000. However, MISNOS technology had viewed IT penetration as significant achievement and not PC proliferation as a figure of merit. Accordingly, factors such as extent of deployment, number of activities for which IT is used, level of IT usage, number of access points for useful applications etc. were to be focused through CICs. However in the absence of MISNOS technology support, these issues could not be addressed even partially. The initial enthusiasm served to boost IT hardware / software sales resulting into new nomenclature like providing citizen interface & e-governance for traditional computerization projects funded by the Government. The primary concern here is non-availability of methodology to reduce ‘cognitive load’ of the end user who is the ultimate beneficiary. Also, there was no software environment for undertaking development activities for Vernacular Computing. The major limitations of ICT are summarized as follows:

(i) The conventional Von Neumann architecture is proving to be inadequate to satisfy the ICT needs for community
information centers population, for introducing e-democracy and similar projects, which are on anvil in coming decades.

(ii) Operating system support for application building in terms of existing software-engineering practices is in the form of several levels of meta abstraction. The existing operating system only partially knows what is really happening in the machine at end user profile level. The context-based operating systems are likely to help to some extent to support software development projects for information technology for masses by deploying MISNOS Technology. Thus, systems that considerably reduce a manageable cognitive load ‘off’ the potential user, is the need of the hour.

A quick journey towards examining the ‘state-of-the art’ off the shelf software development environment indicates that a number of steps like automatic code generators, Computer Aided Software Engineering (CASE) tools etc. have been taken and the most notable of these are the visual environments for programming languages (like Visual C++) where a programmer has been provided ‘drag and drop’ facility for input / output menu development. However, these tools don’t serve the purpose as developers are still forced to follow the ‘iterative modeling’ approach as brought out in the Programming Medium philosophy by Patki (1997-c). The visual programming environments don’t provide facility to assist in concept modeling. Thus Visual Programming is just providing a drafting methodology.

MISNOS Technology is primarily considered as effective vehicle for developing futuristic systems of next generation Information Society promoting Community Informatics including e-democracy. The characteristic features required for futuristic Community Informatics include

- Handling data and information overload at the stage of information modeling
- Adopting design and development philosophy for building systems on the basis of information/knowledge synthesis and not on iterative modeling
- Integrated Vernacular Computing environment instead of mere input / output data entry support in local language

Various technological developments in electronics have resulted in increasing the ‘speed’ of the computer, which is traditionally measured in Million Instructions Per Second (MIPS). The better metric for the computer speed can be in the form the Machine Intelligence Quotient (MIQ) rating as suggested by M. Jamshidi (1997). Each device (irrespective of hardware or software origin) will have its MIQ ratings. The problem in the Information Society is that of the information overload. By information overload we do not mean the insufficient storage capacity on the hard disk, or other storage devices in terms of Mega Bytes. The information overload refers to the phenomena of a typical software application portraying an unmanageable cognitive load, which often dissuades the potential users from using the system. Apart from burdening with unreasonably huge amount of time for understanding the working of the system, such software systems no longer serve the purpose for effective solutions towards Community Informatics.

OBJECT ORIENTED PROGRAMMING SYSTEMS (OOPS) AND CLASS LIBRARIES

Early software developers had no choice and had to develop their software packages using C programming and only tools like SDK were available which were very cumbersome. OOPS brought forth a practice of improving structured programming efficiency. The visual programming also helped to provide an integrated development environment ranging from editing-compiling-linking-executing to version control provision. However, practical minded software professionals feel that the ‘application wizards’ have partially reduced source-coding burden by using automatic code generation feature. It has been revealed
through the usage of C++ class libraries over last two decades that software developers consider the class library as more like a part of project’s source code base and not as a modular reusable component. Hence, the market for class library could never grow in size and the projections that predicted exponential growth in class libraries market using OOPS methodology, have failed miserably. In reality, only software vendor specific class library form Microsoft and similar organizations, could get deployed for improving code generation phase activity. Microsoft Foundation Classes (MFC) is a collection of C++ classes. It can be used to develop Windows programs and considerably reduce the coding effort of the programmer. MFC encapsulates key Windows data structure and have been built using the Operating System (OS)/API functions. It is an attempt to place an object-oriented wrapper around Windows API. Thus, while programming with MFC, one rarely needs to call the Windows API directly. However, it is almost necessary that the programs that can make use of MFC should be written in C++.

The classes in the Microsoft Foundation Class Library (MFC) make up an “application framework” that defines the skeleton of an application and supplies standard user interface implementations that can be placed onto the skeleton. The programmer is required to fill in the rest of the skeleton that are specific to application under development. MFC also has an additional feature for supporting Component Object Module (COM). But still the programs are to be written in C++ only. The individual classes of the MFC are not at a level that persons without in-depth training can use these as components to build full-fledged applications. MFC do not provide any customized support for individual users and without customized support, MFC would not be able to play a major role in making software development easier for masses. The Application Wizards are not substitute in this context. Also, lack of features for Artificial Intelligence in MFC is a bottleneck. The similar lacuna for JAVA have been analyzed and reported by Patki (1997-c).

COMPONENT OBJECT MODEL (COM)

Designing software with components is slightly different from designing with traditional methods. Component Object Model (COM) is efficient method for implementing inter process communication. COM is a group of conventions and supporting libraries that allow interaction between different software components in a consistent, object oriented manner. COM objects can be written in many languages and they can be implemented in dynamic link libraries (DLLs) or in their own executables. With COM, object orientation can be taken to the binary interoperability level. This allows creation of objects compatible at local source level as well as at the application, runtime level.

A client using a COM object need neither be aware about the language in which the object is written nor whether it is running in a DLL (or a separate process). Distributed COM (DCOM) allows objects to be created and accessed on remote computers. DCOM provides a standard protocol that can function on top of any standard networking protocol. DCOM communications also work between dissimilar computer hardware platforms and operating systems. If DCOM has been implemented on both ends of a communication, it does not matter to either the client or the component which operating system is executing the other.

APPLICATION OF FUZZY LOGIC FOR COM/DCOM

The concept used to model reuse in COM is encapsulation and not inheritance. It is stated that COM’s design consideration included two types of inheritances viz. Implementation inheritance and Interface inheritance. While the Implementation inheritance is a mechanism for reusing an existing implementation (i.e. actual implementation behavior is inherited), the Interface inheritance assumes that only the specification of behavior is inherited, enabling polymorphism and COM fully supports it. The implementation inheritance suffers from lacunae of undue coupling between a parent class and child class leading to ‘leak’ violating the encapsulation of the class in an object oriental methodology. Various C++ programming techniques can
lead to the development of reusable binary components. The fuzzy logic based approach helps to overcome the conflict in COM philosophy regarding use of implementation inheritance ‘within’ components versus ‘across’ components. COM restricts use of implementation inheritance for programming ‘within’ components and does not support implementation inheritance ‘across’ components. While the in-depth discussions regarding fuzzy logic as applicable for problem under consideration in this section is beyond the scope of this paper, it would be worthwhile to mention that at ‘class’ level, the data (as protected, private and public) so also the class functions using these data elements, be represented using a fuzzy set with appropriate membership function. Other fuzzy logic operations and fuzzy inference rule base for COM implementation has to be dynamic and not static. The findings of the preliminary work being undertaken in this direction are quite encouraging and are likely to produce a phenomenal impact on the future software development practices for Community Informatics. The existing citizen interface with the Information systems also needs considerable improvements especially when we enter out of the textual information domain. The Query Languages for Multimedia Data on the lines of a unifying framework reported by Nepal et. al. (1999) have potential for deployment in citizen interface applications using multimedia. Authors had discussions with several professionals and it is important to mention here that the general impression regarding usage and applications of fuzzy logic based systems is that it is useful only for ‘Control System’ type of problem solving situations (e.g. Cement Kiln process control, brake systems controller for railway trains, elevators and washing machine control etc.). However, the recent initiatives of applications of fuzzy logic in Information Technology in the form of primitive data types to address complexities arising out of various data structures, FUZOS, File Management etc. brought out in various World Soft Computing Conference Proceedings by Patki et.al. (1996, 1997-a, 1997-b) are definite milestones indicating the road map of fuzzy logic for IT. Fig. 1 depicts Fuzzy Logic based Software Development Model for Community Informatics.

**COM / DCOM EXTENSIONS**

COM is based on interface-based programming and can be used to develop interfaces, which don’t pose any substantial cognitive load on the end user. The Component Object Model (COM) is a component software architecture that allows applications and systems to be built from components supplied by different software vendors is such that it:

- Defines a binary standard for component interoperability
- Is programming language independent
- Is provided on multiple platform

However, with all the above provisions, COM does not provide any support, which can be effectively used to meet the requirements of the information systems need for providing IT for masses. Moreover, the COM objects have limited power to learn as they are being used. Thus, some of the changes that can be incorporated are the incorporation of:

**End User Assistance**

If all the modules of software are made available as individual components to the end user then the customization of software will become a mechanical task. Thus, the cognitive load on the end user will reduce, leading to propagating the use of IT for masses.

**Inculcating the AI methodologies in the COM components**

By implementing the AI methods for searching like heuristic search, best first search, breadth first search, depth first search etc. and the methods for representing predicate logic, frames, semantic nets etc., the functioning of COM components can be made more efficient. This will indirectly improve the prospects of the use of component model based IT software development.

**Platform Independent components**

Until and unless the components developed through COM are made platform independent the actual goal of mass usability would not be
achieved. Although the binary standard of COM does help in making the components portable but without a mechanism similar to byte codes of JAVA is developed, the goal of total portability cannot be realized.

**Support for Mobile Computing through Cell Phones**

Mobile Computing is an upcoming field and the recent spurt in the sales of the mobile phones in developing countries like India presents an opportunity for setting up new types of Community Informatics Centers. Thus, it is very important that COM provides a support for Mobile Computing in the form of standard for WML, WAP and XML that are specific to the mobile environment.

**CONCLUSIONS**

Software development for Community Informatics is an emerging discipline. The object-oriented technology had resulted into foundation classes e.g. MFC in Visual C++ studio environment. This has created a trend amongst programmers and system developers towards remembering dictionary words for creating sentences to write essays. This similarity is noteworthy when we consider software systems development for Community Informatics. The existing COM/DCOM support lacks essential features to create application software using software components to reduce the cognitive load of the end user. It is essential that fuzzy logic based interfaces are required to be developed along with predicate logic support (similar to Prolog) for COM type of methodology to serve as a useful career for programmers. The general trend / practices adopted by hardware/software developers for Community Information Centers, seem to indicate a saturated plateau for the existing systems and the feedback indicate that unless new technologies (like MISNOS) are developed and deployed, we shall be no where near the goals of creating and implementing e-democracy projects.

**ACKNOWLEDGEMENTS**

Authors acknowledge the support and environment provided by Shri S. Lakshminarayanan, Additional Secretary, Department of Information Technology, for undertaking the study exercise on this topic in a short span of time. Focused discussions with Dr. S.P. Nawathe, Scientist ‘F’, Mr. S. Sivasubramanian, Scientist ‘E’ and Ms. Tulika, Scientist ‘D’ and Mr. S.S.Sarma, Scientist ‘B’ of DIT have been useful. The interactions with industry representatives were enlightening and thought provoking.

**REFERENCES**


Fig. 1. Futuristic Software Model for Community Informatics