Investigating Intelligent Workflow and Intelligent Agents in Software Flexibility

Shokooh Abedi Dehkordi\textsuperscript{1} \\
e-mail: Shokooh_abedi_2008@yahoo.com

Hossein Monem\textsuperscript{2} \\
e-mail: ali.monem@gmail.com

Author(s) Contact Details:
\textsuperscript{1,2} Islamic Azad University, Fars Science and Research Branch, Shiraz, Iran

Abstract—Developing software systems are easier to design, cheaper to maintain, and more robust therefor software engineering has been seen as a continuous quest in developing software systems. Recently, most major software companies have either acquired or developed workflow components for integration into their existing software platforms. Software flexibility shows that indicate if the software is easy to change. Workflow management by modeling business processes explicitly and managing business processes as data helps achieve software flexibility that they are much easier to modify than conventional program modules, while workflow systems must be adaptive in order to effectively support today’s dynamic, uncertain, and error-prone collaborative work environments. In a workflow environment an innovative technique of achieving software flexibility was investigated to achieve a robust, cost effective and flexible software. Software can be made more robust, more cost effective to maintain, and easier to change by incorporating workflow technology and intelligent agent techniques into modern information systems. In this paper two steps investigation consist of first: critical and deep investigation on the software flexibility issues, and second, an investigation on the integrated system architecture that include of workflow coordination mechanisms and the agent-based decision making capabilities, more insight was prepared for further researches in this disciplines.

Keywords – Intelligent Workflow Techniques · Software Flexibility · Intelligent Workflow Management · Intelligent Workflow System

1. INTRODUCTION

The central role of flexibility in software design and implementation in software design is recognized by classic and contemporary literature. Structured design, modular design, object-oriented design, software architecture, design patterns, and component-based software engineering, play the main role in software flexibility. Software flexibility has effect on maintenance cost it is capable of handling certain exceptions in an automated manner.

Two problems in the current notion of ‘software flexibility’ are: The absence of reliable metrics and the second is misconceived that is as an absolute quality. The definition of software flexibility in IEEE is: The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed [20]

However, what is software flexibility and how to measure it aren't be clearly defined. Although intuitively software flexibility is understood as the ability to respond to changes, and is occasionally used for evaluating the quality of software, but there isn't yet a good definition for software flexibility because:

First, the “flexibility” is a word with rich meanings. It is sometimes used interchangeably with other terms, such as adaptability, plasticity, elasticity, agility and versatility, etc. Second, to the author’s knowledge, there are no well-defined and unified software flexibility measurement criteria. Third, software flexibility, similar to material flexibility, is one of software’s internal properties. Though, how easy the software is to use, how long it takes to make changes on
existing software, etc., to some degree reflect software flexibility properties, such external behaviors do not reveal the essence of software flexibility [16].

Software engineering has been seen as a continuous quest of developing software systems that are easier to design, cheaper to maintain, and more robust [14]. Data-oriented software architecture, object-oriented language, component-based software development, and document-based interoperability have been achieved in software engineering. Data independence and process independence are more important than others, when the data structures modified, data independence became a measure of the robustness of business applications. Relational database became dominant in the database market because of the achievement of significant data independence. When the process model redesigned, process independence became a measure of the robustness of business applications. In the last few years the drive towards more process independence has led to the proliferation of workflow systems in the software industry. Very recently, most major software companies have either acquired or developed workflow components for integration into their existing software platforms. Workflow management by modeling business processes explicitly and managing business processes as data helps achieve software flexibility they are much easier to modify than conventional program modules. Workflow management systems enable reuse of process templates, robust integration of enterprise applications, and flexible coordination of human agents and teams. In this paper, it is studied additional techniques based on intelligent agents to incorporate more flexibility in conventional workflow management systems. The research goal is to further improve software flexibility by achieving more process independence [4].

2. LITERATURE REVIEW

A. Software Flexibility

Issues in the management of changing business processes from strategic, operational, and system perspectives have been studied by researchers [2, 6, 8]. Process changes can occur often. Many business processes are not well understood and difficult to formalize, leading to mismatches between the system specifications and real needs of the business. In addition, there are often exceptions to the basic routines of the business, requiring the system to be robust in order to handle process deviation from the norm [4].

Technology flexibility is defined as the characteristics of technology that allow or enable adjustments or other changes to the business process [18]. They proposed a measurement framework for technology flexibility that include such factors as modularity, change acceptance, and consistency in the structural flexibility dimension, and rate of response, expertise, and coordination of actions in the process flexibility dimension.

System adaptability and system versatility are two software flexibility concepts [21]. System adaptability is the capability to modify the system to cope with major changes in business processes with little or no interruption to business operations. System versatility (or system robustness) is the capability of the system to allow flexible procedures to deal with exceptions in processes and procedures.

In this paper, the concepts of software flexibility and apply the results to the development of flexible process technology is extended and unify by means of intelligent workflow techniques.

B. Flexible Workflow Systems

According to the Workflow Management Coalition (WFMC), a Workflow management system is “a system that completely defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic.” [19]
Researchers in workflow technology and management are concerned with developing flexible workflow systems. Klein, Dellarocas, and Bernstein (2000) claimed that “Workflow systems must be adaptive in order to effectively support today’s dynamic, uncertain, and error-prone collaborative work environments. … Workflow systems currently provide little support for adaptive processes. Most do not allow one to modify a process model once it has started executing. Exceptions are handled by attempting to include the workflow pre-defined conditional branches for all possible contingencies.”

Researchers pursue several important directions to achieve workflow flexibility:

• Exception handling: Fault tolerance is a key requirement in process support systems including workflow systems. Hagen and Alonso (2000) presented a solution for implementing more reliable processes by using an exception handling technique as used in programming languages.

• Dynamic model evolution: Dynamic workflow change requires the management of changing process models. For instance, Ellis and Keddara (2000) addressed the problem of unambiguously specifying process model changes, via taxonomy of change modalities and a language for the unambiguous specification of procedural change.

• Emergent process modeling: Another general approach for dealing with dynamic adaptation is based on partially specified process models and depends on flexible enactment systems to refine and execute them at the runtime [15, 9].

In this paper, software flexibility is built upon the basic ideas of dynamic workflow management and proposed intelligent workflow techniques.

C. Intelligent Agent perceptions

Intelligent agent technology is one of the fastest growing areas of research and system development in IT [11]. Intelligent agents and multi-agent systems represent a new way of modeling many complex information management and decision tasks. Agents also represent a new computing environment for designing and implementing complex software systems [12].

Intelligent agents refer to autonomous software entities that can navigate heterogeneous computing environments and can, either alone or working with other agents, achieve user-delegated goals.

3. Forming Software Flexibility

Software flexibility conceptualized in two levels, i.e., technology flexibility and system flexibility. In software flexibility, is referred to the ability of a software application to deal with exceptions to the process model at the runtime and to cope with periodic changes to the process model. By technology flexibility, Software flexibility can be achieved by adopting the right combination of specific modeling paradigm, programming technique, software architecture, and development methodology. This level of flexibility is analogous to the combination of the structural and process dimensions proposed by Nelson et al. (1997), but with less emphasis on the organizational effects. By system flexibility, both system adaptability and system versatility were discussed by Zhao (1998).

The term of hierarchical process management is coined. In a large-scale information system such as a B2B transaction processing system, the overall business process consists of many sub processes, including inter-enterprise processes, intra-enterprise processes, inter-function processes, and intra-function processes. It is referred to the modeling and execution of this process hierarchy as hierarchical process management. In the next section, it is shown that different information techniques such as workflow and intelligent agents can be applied to improve the technology flexibility and system flexibility at the various levels of the process hierarchy [4].
4. Intelligent Workflow Organization

A. Workflow-Based Flexibility

Workflow technology is becoming the base of the next generation of enterprise application systems because it enables better flexibility of software systems in several respects: consist of first, specialization of components into functional and process components: The workflow model essentially captures the inter-component processes, using a component-based software development philosophy. Second, in some ways, a regular component such as a database can be seen as a functional component, while the workflow engine can be considered as a process component. This is the essence of hierarchical process management. Third, the separation of process design and process execution: The workflow paradigm enables the separation of process design from process execution by managing a process model as a template that can be instantiated at the runtime. This makes it possible to isolate the change of processes to different levels of the process hierarchy, resulting in reduced process interdependence. As a result, changes to processes can be made faster and cheaper. Forth, interoperability between process components: Given two workflow-coordinated enterprise systems, the interoperability between the two enterprise systems is turned into the interoperability between two workflow engines. This helps improve the speed of establishing, redesigning, and executing the inter-enterprise processes, thus making the overall system more flexible when the business relationship between two enterprises undergoing changes [4].

B. “Intelligent Agent”-Based Flexibility

Intelligent agent techniques can be applied to improve the flexibility of information systems because of its several inherent features which claimed by Zeng and Zhao (2002) and summarized in this part. These features include of first, specialization that means intelligent agents can be programmed to follow special instructions with a specific goal from a generic architecture. This enables fast building of software components. Second, negotiation, that means intelligent agents can negotiate with one another to exhibit versatile behaviors in the system that would not be possible through rigid procedures. And last one is coordination that means intelligent agents can be used to build a collaborative community of agents (including both human and software agents) that is more flexible than the typical workflow model.

C. Flexibility via Intelligent Workflow

It is proposed to combine the workflow-based flexibility and the agent-based flexibility to achieve significantly improved overall software flexibility. It is referred to this endeavor as flexibility via intelligent workflow. This approach has been also expressed by Zeng and Zhao (2002) as following properties.

- Inter-enterprise flexibility can be achieved by interoperating workflow engines that each coordinates an intra-enterprise business process.

- Intra-enterprise flexibility can be realized by means of the workflow management paradigm as aforementioned.

- Inter-component flexibility is made possible by means of both workflow management and agent-based components.

The next subsection illustrates how these three types of flexibility are achieved under the hierarchical process management paradigm we propose in this paper.
**D. Hierarchical Process Management**

Figure 1 illustrates how workflow engines are used to link business processes between two enterprises. The interoperation between two enterprises requires a collection of such activities as advertisement of services, negotiation of contracts, and outsourcing of processes, which has been referred to as cross-flow [17].

![Figure 1: Inter-enterprise process management (adopted from Zeng and Zhao (2002))](image1)

Figure 2 shows how to manage inter-department processes by modeling them under one or more workflow systems. An inter-department process is more stable than an inter-enterprise process because the relative stability of relationships and the close coordination and planning ability within the same enterprise [4].

![Figure 2: Inter-department process management (adopted from Zeng and Zhao (2002))](image2)

Inter-component process can be managed by integrating workflow components with intelligent agent components [4]. As shown in Figure 3, the coordination process among a community of intelligent agents is more flexible than a regular workflow coordination process because the interactions between the coordinating agent and the task agents are achieved through flexible negotiation in a distributed manner based on a common agent language and program interface.

![Figure 3: Inter-component process management (adopted from Zeng and Zhao (2002))](image3)
5. A General Background for Evaluating Software Architectures

Software maintenance cost often constitutes a significant part of the total cost associated with developing and operating a software system during its life span. If a software system is more flexible, it can reduce the maintenance cost then Software flexibility through several ways has major impact on software maintenance cost has major impact on software maintenance cost. They are consist of: First, a software system is capable of handling certain exceptions in an automated manner then it developed to be flexible, which in turn cuts back the effort and resources that have to be invested to deal with these exceptions by both the software users and the system development and maintenance personnel. Second, in today’s competitive marketplace, an enterprise typically goes through frequent changes in its ways of dealing with their customers and suppliers. A flexible software system provides a viable alternative in dealing with these types of software maintenance costs since it is easier and more cost-effective to reconfigure and reassemble software components to meet changes in user requirements. Three possible system architectures are considered in this study: a) a conventional information system in which no workflow or intelligent agent techniques will be used, b) a workflow-based information system, and c) an intelligent workflow system that combines workflow and intelligent agent techniques [4].

6. CONCLUSION

The benefits of flexibility are not only intangible but also difficult to quantify, so more quantitative approaches require evaluate and improve software flexibility. In this paper, some techniques of achieving software flexibility in a workflow environment was investigated that enhanced by intelligent agent technology.

A general evaluation framework was presented with respect to total software cost which can be used to analyze the economic impacts of various types of software architectures. Some general technological and managerial insights are offered as to the varying degrees of software flexibility and cost implications for conventional software, workflow systems, and the proposed intelligent workflow systems. The general evaluation framework can also serve as a theoretical basis for making potentially critical software architecture decisions facing virtually all IT adopters [4].

REFERENCES


[16] Limin Shen, Shangping Ren , “Analysis and measurement of software flexibility based on flexible points” , Published in the Proceedings of Smef – 2006


