Structured Information Flow (SIF) Framework for Automating End-to-End Information Flow for Large Organizations

Shishir Bhatia

Thesis submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Approved:

James D. Arthur, Chair

Calvin J. Ribbens | Osman Balci

February, 2006
Blacksburg, Virginia

Keywords: End-to-End Information Flow, Paperless Office, Information Flow Framework, Information Flow Automation Model, Electronic Information Flow

Structured Information Flow (SIF) Framework for Automating End-to-End Information Flow for Large Organizations

Shishir Bhatia

(Abstract)

For almost five decades, since the advent of the first computers for commercial use, the dream of the Paperless Office, a.k.a. total Information flow automation, has eluded the industry. Now, with the emergence of Internet- and Web-based technologies, daily we see examples of organizations like eBay and Amazon that perform their business in a fully automated manner without the use of paper or pen. However, bigger and older organizations that have more complex functions, like government organizations, have not been very successful in harnessing the latest technological innovations to completely automate their Information flow. We propose a Structured Information Flow (SIF) framework that provides the conceptual infrastructure to automate small and big, new and old organizations alike. The ease of the transformation is due to three key features of SIF that set it apart from any other Information flow automation scheme.

First, SIF utilizes the attributes of the organization, such as the existing reporting structure, to model the automated Information flow. The rules governing the flow of Information are based on the hierarchy already in place, for example: *A senior can view any Information owned by any of his/her direct subordinates*. Second, SIF characterizes external organization entities as a special case of internal organization entities, allowing for seamless integration of the Information flow to and from them. Third, the SIF framework is independent of platform, method, organization, or technology. This gives it a generic nature that makes it applicable as a platform to implement multiple types of automated e-systems such as e-commerce, e-education, e-training, e-governance, etc.
In this body of work, we formally define the SIF framework using state transformation language and a visual representation scheme specifically developed for this purpose. We also define the Information Interfaces, which are the mechanism for implementing rules- and constraint-based Information flow in SIF.

ACKNOWLEDGEMENTS

I am forever thankful for the wisdom and guidance my advisor, Dr. James D. Arthur, has provided me over the past three years. It is because of his patience, support, and encouragement that I have finally completed this thesis.

I would like to thank my parents, who have always been there for me, lifting my morale through hard times.

Last but not least, my deepest love and thanks to my newly wed wife, Mini, who stood by me as I worked long hours towards finishing this work.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 – Introduction

1 Introduction

Over the last five decades, virtually everything in our society has evolved in a way almost unforeseen in 1950. Communication, politics, fashion, entertainment, social norms – nearly everything that shapes our lifestyle has undergone radical change and tremendous progress. The overall result is great improvement in convenience and capability for the masses. However, the traditional office has, in most cases, remained exactly that – traditional. While office processes have been greatly enhanced by modern technologies, they still fundamentally rely on an old-fashioned paper-based flow and tracking of Information. The emergence of mainframe computers and networks for commercial use in the late 1950s [Grudin 1990] and the development of microcomputers in the 1970s [Clement 1978] promised to revolutionize the way organizations worked. By the 1980s, the newest innovations in computer technology promised a paperless office using electronic Information flow [Liu 2000]. Electronic communication in place of paper-based communication guaranteed efficient organizational processes because it afforded a great many advantages, such as requiring less handling, providing fewer opportunities for human error, automatically creating a tracking route, and facilitating communication between various sites, among many others [Petrakis 2000]. However, to date, we do not see many organizations that have totally embraced the concept of a paperless office. This fact is in high contrast with the soaring Information technology (IT) expenditures of most organizations. A survey in 1997 showed that, in the surveyed organizations, 94% of Information is still recorded on paper with an estimated 2.4 billion new sheets placed in folders each day [Liu 2000]! This situation begs the question: *why do we have so much paper floating around?* If we have IT infrastructure, and electronic documents, and trained, dedicated personnel, why do we not fully embrace the paperless concept?

We conjecture: the problem lies in the approach that most organizations utilize for Information dissemination using IT systems. In most cases, the Information flow is one-dimensional, like emails, from source to destination. The Information flow in most organizations is not discriminatory – it does not categorize the Information and the destination to automatically decide if certain Information should be delivered to a certain destination or not. If we want to even get close to the dream of a paperless office, we need a standard framework to structure the Information flow automatically. This includes not only structuring the Information flow within the organization, but the incoming and outgoing data as well. In this paper, we present our approach, which discriminates based upon the Information type and the destination type to determine valid Information flow paths. This approach can be used to completely automate the Information flow of an organization.

1.1 Motivation

When computers first made their way into industry, they promised a revolution that would change the way most organizations work. There is no doubt that the past two decades have witnessed tremendous change in the way day-to-day work is carried out in organizations using computers, but have these changes been at par with the expectations? We are still far from the total automation of Information flow that will help us fulfill the dream of the paperless office. What is most required at this point is a simple, standard plan that provides a universal framework for converting the traditional into the state-of-the-art. This paper details just such a plan, and the main points it entails are explained below.

1.1.1 Information Flow Automation

Most organizations today use personal computers (PCs) to aid daily work, but one is hard-pressed to find an organization that fully utilizes IT to facilitate or automate each phase of their process. Rarely will we find, for example, a government municipal organization utilizing IT solutions for end-to-end fulfillment of an incoming complaint, such as a leaking sewage problem. A plan for converting these organizations to paperless environments must be easily applicable to large, complex operations. Why are only the more recent organizations like Amazon.com or Ebay able to completely automate their end-to-end processes using Information flow? The answer is in the fact that these organizations were conceived after the Internet revolution and, hence, had the advantage of starting from a current perspective when creating their Information flow infrastructure.

There was no history of doing things the "old" way, so these companies started with a clean slate, designing their processes based on the current technology and all of its capabilities. More established entities can be crippled by the paradigm of the traditional paper flow, so they are tasked with not only updating their equipment and revising their procedures, but also reinventing the way they view Information handling. Does this mean that organizations that matured prior to the Internet revolution cannot reap the full benefit of Internet-based IT solutions to automate their Information flow? No, we conjecture that these organizations can also fully automate their Information flow using Internet-based technologies. However, older establishments, such as government organizations, have more evolved and complex internal processes as compared to eCommerce processes, which are primarily search, select, buy, and confirm. In this work, our primary focus will be to define a framework for Information flow automation of large organizations like government agencies.

1.1.2 Standard Information Flow Automation Framework

Perhaps the single biggest factor responsible for keeping large organizations from Information flow automation is the lack of a commonly accepted standard for Information flow automation. Until now, every effort to automate Information flow in large organizations has been in isolation from other similar efforts. There has been no attempt to unify or standardize the approaches, so no one entity has been able to learn from the successes or failures of others. Without contributions to the collective base of knowledge in the industry, there has been no substantial growth in common knowledge about the best paths to paperless conversion. A standard approach for automating the Information flow, such as the one offered in this paper, will serve as the vehicle to capture the refinement that the science of automation will grow through with every effort. Such an approach should be general enough to be applicable to almost any organization, and should have an open-ended architecture that could be implemented using any appropriate technological platform. In short, the suggested framework for Information flow should be organization-independent and technical platform-independent as well.

1.1.3 Criteria for the Standard Framework

In addition to the above high-level requirements, at a deeper level, the framework should be able to allow for discrimination of incoming and outgoing Information flow, as well as that retained within the internal loop. To support such discriminatory Information flow, the framework must accomplish, at a minimum, three tasks. First, it must categorize types of Information, such as restricted or unrestricted. Second, it must also categorize the roles in the organization in a hierarchical manner. Third, the framework must also define a set of rules that will govern the association (access paths) between the Information and the roles.

These three criteria illustrate the basic framework of our conversion plan. However, for it to be applicable to the large, older organization, it must also be flexible enough to embrace the organization's current practices without imposing a new set of alien processes. In short, the process changes imposed on the organization by the Information flow framework should be minimum to make it friendlier and compliant with the older organization.

To propose such a framework is the intention of this work.

1.2 Problem

An examination of the history of IT development helps to illustrate the course by which some organizations, it seems, were left behind. By the end of the 1960s, the huge mainframe computers that provided big organizations with the much-needed horsepower for data processing were still restricted to only the largest corporate and government organizations. A decade or two later, PCs were finally beginning to make their way to the general public. Over the next twenty years, PCs improved exponentially and paved the way for networked communication capabilities. Many organizations embraced PCs and networks to aid and automate their internal functions. But even as late as the mid 1990s, the average household did not have either a PC or network access. Then came the age of the Internet and cheap computer hardware. PCs were finally becoming common in traditional households and empowered the masses to access Information instantaneously and at will. Quickly, it became an expectation that organizations not only provide Information through the Internet, but also allow customers to interface and receive dynamic and personalized responses. Many organizations embraced the public's desire for online services like bill payment, account queries, and placing orders. The challenge was that most organizations with costly intranet installations were not geared to interface with customers at multiple points using the Internet. Additionally, the internal processes of the organizations were not poised for customer-driven input via the Internet. Therefore, in this seemingly IT-automated structure, the basic connection of processes still had to be performed 'by hand' – negating the potential advantages of the technological capability.

Conversely, however, the organizations that were born in the post-Internet era utilized the IT solutions to not only receive the input from the customer (via Internet), but also to set up the automated internal processes that are driven by the customer input via Internet. For example, Ebay receives a customer request to purchase an item electronically, which then prompts the automated system to process the order. To close the loop, the output of the process, like acknowledgement of a shipped item, is triggered automatically and electronically transmitted. The key to fully automating the Information flow is the seamless integration of the organization's internal processes in response to external requests. This is difficult to achieve for organizations that have, for all of their history, been using conventional methods for receiving inputs (e.g.: manual data entry), processing, and providing output. Various recent methodologies to fully automate organizations call for radical changes in organizational functioning; most organizations, for monetary and cultural reasons, have not been able or perhaps willing to struggle with this daunting task [Smart 1995]. However, an approach that would help preserve the best part of an organization's conceptual structure, while providing seamless integration of the input, processing, and output, would be far less daunting and even easily implemented.

1.2.1 Typical Information Flow Vs. Automated Information Flow

A simplified end-to-end Information flow view for a typical large organization can be broken down into five simple steps, as described in Figure 1.

Step 1. Customer enters Information, typically in a paper form at one of the local offices of the organization. Let's say this item of Information is a user complaint form for an energy provider.

Step 2. The complaint form is manually received and routed to the appropriate role.

Step 3. A combination of a number of steps now takes place while the Information is processed and the complaint is attended.

Step 4. The output of the processing is sent to the appropriate department in the organization for communicating to the customer.

Step 5. Customer receives the output.

Clearly, this is not a completely automated process, even though portions of step 4 can be semi-automated. Such processes require a significant amount of time and effort, and are prone to errors, too.

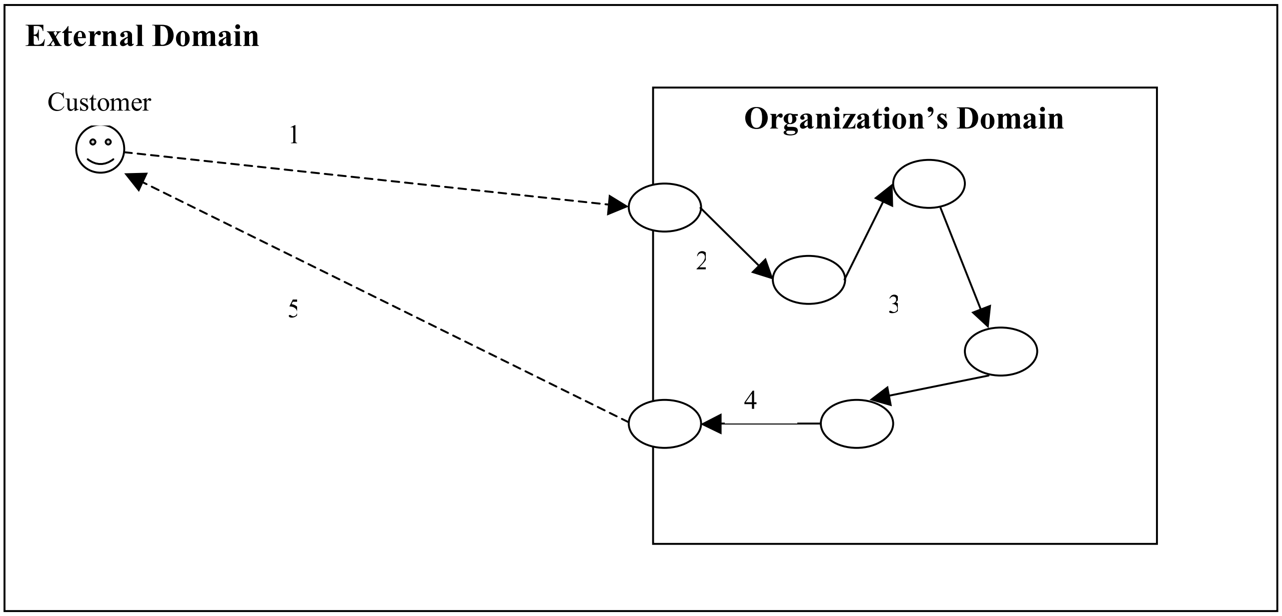


Figure 1.1: Generalized Information Processing View of a Large Organization

By contrast, seamless automated processing should have consistent, prescribed processes, each triggering the next, as shown in Figure 1.2. It consists of Steps 1 through 3.

Step 1. Customer populates a Web-based form to register the complaint.

Step 2. The complaint is saved in a centrally accessible database, and the appropriate roles in the organization receive the notification to act on it.

Step 3. As soon as the status of the complaint is marked 'Closed,' an automated output is generated for the customer.

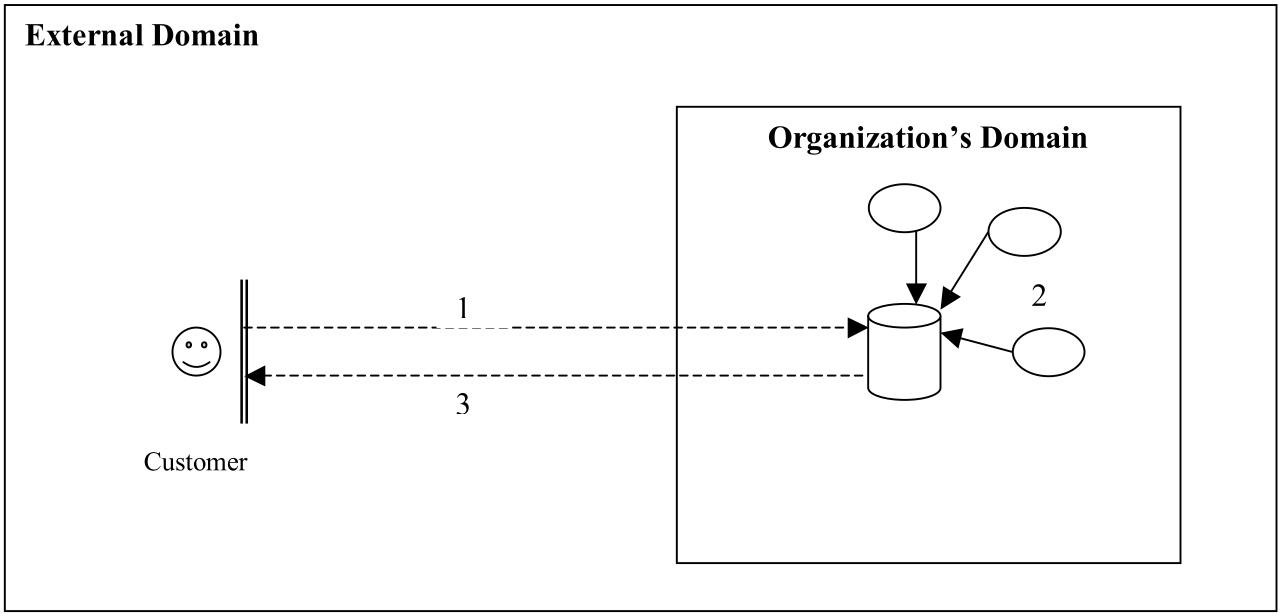


Figure 1.2: Generalized Automated Information Processing View of a Large Organization

What makes it a challenge for older organizations that work in the manner depicted in Figure 1.1 to perform as depicted in Figure 1.2?

The problem is not in the IT tools that are available to them. After all, most organizations have achieved a level of IT integration that can be made to support limited interactions with the external domain. The problem is in the lack of an Information Flow Framework to guide the implementation of a system that would seamlessly integrate the inputs and outputs to and from the external domain with the internal processing of the organization.

1.3 Issues

At a very high level, the way industry uses IT systems for performing internal processes is by storing Information into single or distributed databases and using reporting tools, emails, File Transfer Protocol (FTP), etc., to disseminate Information for processing. However, most of these systems are 'dumb' [Bayer 1996]; meaning they just hold the Information but do not decide who should have access. In other words, they are not intelligent enough to look at the Information piece and decide which roles should be given access to that Information piece to finish their work. Our observation is that only a system intelligent enough to dynamically decide which role should get access to Information to fulfill the business process can perform end-to-end seamless automation. Only such a system can dynamically register the Information received from external domain entities, allot permissions to internal domain entities enabling them to work on it, and ultimately send the output back – making Input, Processing, and Output seamless.

Designing such a generic Information flow framework is a multidimensional problem. It requires a very well-thought-out underlying Information flow framework standard, which is generic enough to accommodate the Information flow for the day-to-day needs of most organizations, and not just target e-commerce or e-business-related organizations. We put forward a list of issues below that must be overcome by the Information flow framework alluded to above.

1. Intelligent Routing of Information: The framework should provide an underlying mechanism that will allow the system to intelligently route the Information. This mechanism will include the rules that will determine which roles should get access, and to what extent, to the newly registered Information piece. This will allow the system to determine whether the request is a valid one or not in real time when an Information access request is made.

2. Reflecting the Organization's Structure: The framework should be generic enough to suit most organizations, and at the same it should not introduce processes alien to the organization. In other words, the framework should lend itself to the organization rather than dictating terms to the organization.

3. Organization and Technology Independent: The framework, in order to be widely accepted and used and, hence, retain learning from a variety of organizations, should be organization and technology independent.

4. Pragmatic: The framework should provide a pragmatic approach to Information flow. The framework should be intuitive and allow the organization to be abstracted at different levels for easy representation.

We conjecture that an Information flow framework that accommodates all the above issues will be ideal for structuring the Information flow of almost any organization, which will lay a strong foundation for achieving total Information flow automation.

1.4 Solution Approach

In this paper, we propose the Structured Information Flow (SIF) framework. SIF is a framework that structures the end-to-end Information flow of an organization, from initial input to processing to the final output. It utilizes the characteristics of the organization and its components to structure the Information flow in a manner that seamlessly integrates input from and output to the external domain with the organization's internal processes.

The following capabilities built into the SIF framework enable it to overcome the four issues mentioned in the previous section:

1. Seamless Integration: SIF treats external domain entities as a special case of internal domain entities. Within SIF, there are rights and privileges associated with the entities. External domain entities are the same as internal domain entities, but with a minimum set of rights and privileges. This allows external domain entities to register (input) Information and receive Information from the organization in a more coherent manner.

2. Rules-Based Flow: Once the Information is registered or created by internal domain entities, there are rules and constraints that govern the access and, subsequently, the flow of Information in and outside the system. The rules utilize a) characteristics of the Information and b) characteristics of the entity trying to access the Information to govern the flow of Information.

3. Reflecting the Organizational Structure: SIF is applicable only on hierarchical or semi-hierarchically structured organizations. It uses the senior- subordinate relationship between the entities and the Information categorization, discussed in the next point, as the basis for some of the Information flow rules. One such rule, for example, is that a senior entity can view a subordinate's Information at any time. The rest of the Information flow rules are based on the Ownership of the Information. That is, the entity that owns the Information has multiple rights over the Information and could extend those rights as well. These rules are reflective of the natural Information flow in the organization. Hence, the SIF approach induces the least amount of deviation from the natural way things work in most organizations.

4. Information Labeling: There are two types of Information in an organization – restricted and unrestricted. SIF labels Information as Restricted - with restricted rule-based access, or Universal - free for all access. Labeling the Information is key to the automated Information flow. SIF utilizes the hierarchical standing of the entity as well as the ownership right vs. the Information label to determine whether particular Information access is valid or not.

5. Organizational and Technical Platform Independent: SIF is a generic and technology-independent framework for automating Information flow. As it is applicable to all organizations with hierarchical structure, it can act as a vehicle to capture the learning from one organizational implementation to another.

6. Pragmatic Approach: SIF uses abstraction at different levels for the ease of Information dissemination. For example, various entities abstract together to form a functional group, and SIF provides a logical way to aggregate the Information at the functional group level, making it easier to search and access Information.

The coming chapters will describe how SIF incorporates the above six capabilities. Chapter 3 introduces the logical components of SIF. Chapter 4 formally describes the SIF model using notational language and state-transition diagrams. Chapter 5 describes the requirements of the proposed Information Interfaces, which are a mechanism to implement the rules-based Information flow proposed in SIF.

Chapter 2 – Background

2 Background

This chapter presents a survey of past work in the area of Information Flow Automation. In this chapter, we also analyze why the previous approaches failed to provide a generic Information flow framework that could serve as an industry-wide standard.

2.1 Information Flow Process

Generally speaking, Information is defined as: 'Knowledge derived from study, experience, or instruction' [http://dictionary.reference.com, 2005].

In a computer science context, Information is defined as: 'Processed, stored, or transmitted data' [http://dictionary.reference.com, 2005].

The terms *data* and *Information* are frequently confused. Max and Augusti define *Information* as that component or portion of *data* that can be used by users for some specific business or scientific purpose [Boisot 2004]. For the same reason, *data* can be freely distributed, while *Information* is deeply and safely encrypted in it using various keys. It is important to understand this distinction as we endeavor to solve the *Information flow* problem, which is not to be confused with a *data flow* problem. Networks and operating systems (OS) deal with *data flow* in the form of bits and bytes, while the *Information flow* of concern here is that portion of data that is of particular importance for the functioning of businesses and organizations. Information is encrypted into data to be transmitted over the network using the hardware infrastructure of servers, network cables, routers, etc. Hence, no networked Information flow excludes data flow. As depicted in Figure 2.1, at a granular level, Information flow is comprised of data flow.

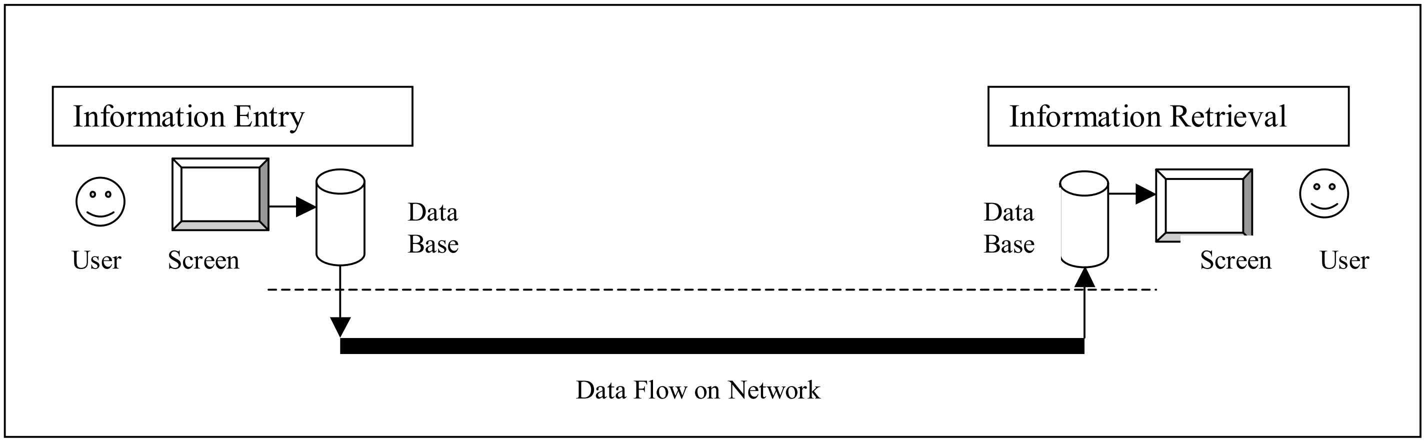


Figure 2.1: Information Flow Process

2.2 Current Attempts at Information Flow Automation

This explication of Information Flow leads now to an analysis of previous attempts to automate it.

It can be argued that one of the first efforts in the pre-Internet period to structure Information flow on a networked environment by controlling user access is the Unix File System [http://www.unix-manuals.com 2005]. Unix offers a certain amount of Information control by assigning each Information file a security label, and only certain users are granted controlled access (Read, Write and/or Delete) to the Information. This is a simple yet robust, time-tested way of ensuring that only the appropriate people ever touch the Information, yet it has several obvious drawbacks that prohibit it from being used as an Information Flow Automation framework.

- It has no obvious way of accounting for the organization's structure and complex hierarchical relationships between people.
- It does not support seamless integration with the masses over the Internet.
- It is not simple enough to be used by a businessperson without advanced training.

These issues combine to present an over-simplified but accurate explanation of why Unix OS capabilities are not sufficient to automate the Information Flow. While

Unix OS is perfectly acceptable for managing data flow, we must conclude that the Information flow problem has to be solved above the OS layer. In fact, there are already architectures like J2EE and .NET that reside on top of OS and promise segregation of technical details from the business user by defining different Tiers: User Tier, Container Tier, and the EIL Tier [Hammouda 2002]. The business user interfaces only with the User Tier without seeing the intricacies of the implementation. However, a robust OS with a well-defined networked data flow combined with an Internet deployment engine like J2EE still does not provide a logical means of going entirely paperless without a well-defined Information Flow Structuring framework. In the section below, we examine a few prior attempts at Information flow automation and the reasons they failed to become a generally accepted framework.

2.3 Prior Attempts at Information Flow Automation

The efforts in Information flow automation can be broadly categorized into the Pre-Internet era before the 1990s, and the Post-Internet era towards the late 1990s.

Before the Internet era, the quest for total Information flow automation was fueled by the desire for the paperless office [Patterson 1981]. Exciting new developments in computer technology over a very short period of time made the notion of a paperless office appear to be an achievable dream. By the end of the 1980s through the mid 1990s, there were a number of efforts that aimed at automating Information flow in the office using various strategies to devise a paperless office.

The Quest for a Perfect Documentation System

Diana Patterson and George Olah in 1981 put forward the specifications for a ‘perfect’ documentation system that focused on gearing the office to a paperless environment [Patterson 1981]. The emphasis of this effort was on issues around indexing Information for fast rendering on the screen, scanning a screen full of Information, using light pens to enter the Information, etc. Interestingly, this effort struggled to work around technical limitations like storage space, data retrieval speed, and data entering speed.

Such limitations no longer exist or are greatly ameliorated for an effort this size, negating the solutions proposed by this body of work.

Document Image Processing (DIP) Initiative

Another effort in the early 1990s by T.T. Lau of the Hewlett-Packard Lab employed Document Image Processing (DIP) as the vehicle to achieve a paperless office [Lau 1990]. In his words: "*DIP involves scanning and electronically storing the documents so that they can be retrieved and used without needing the paper copies or steel file cabinets*." The major flaw in this approach is that there must be an original paper document, which can then be scanned and used for various purposes. The technology at that time did not allow a remote client side form that could be completed by the end-user/customer and, hence, eliminate paper in totality. Given the available technology when this method was proposed, it was a reasonable approach; but certainly today the average office is equipped for much greater efficiency.

2.4 Why Pre-Internet Efforts at Information Automation Failed

What we can conclude from this discussion of prior efforts at Information automation is that the innovative solutions are built within the boundaries of technical feasibility. Twenty years ago, the concept of true automated Information flow was simply not feasible to implement. An end-to-end connection with the user or the ultimate customer could not then be established, but is now possible with widespread and deeply penetrated Internet use.

Other major factors that contributed to the failure of automated Information flow efforts in the pre-Internet era are:

1. Computer hardware was very expensive, leading to restricted computer and network access.

2. In the rapidly changing technological scenery, there was an absence of a time-tested and stable technical platform on which to implement the automation solution.
3. Because of the expensive hardware and lack of network access to the public, the researchers did not include the Information to and from the customer in the scope of the automation.

By the mid 1990s, in the pre-Internet era, researchers realized that the concept of a completely paperless office was a little premature. They took a step back to aim at a reduced-paper office instead [Smart 1995].

2.5 The Post-Internet Era

The post-Internet era is the era of Web-based business, education, commerce, etc. The Web has emerged as a perfect carrier of Information from the ultimate customer into the organization and vice versa. The stage is set for the emergence of a truly paperless office, and while some newer organizations have achieved that state, still there is no model by which other organizations can adjust their practices. Even attempting to imitate a Web-based business to automate Information flow would be pointless because these individual systems are limited by their design. Because of the immense profit potential for e-businesses, most efforts at utilizing the Web for Information flow are based on instant gratification or quick moneymaking. Most of the dot-com organizations are geared towards receiving order and payment from the customer and then fulfilling the order request through semi-automated processes. The focus on rapidly harnessing the market penetration through the Internet has virtually shadowed the development of a broad-based solution for Information flow that seamlessly integrates an organization's internal flows with the flow of Information to and from the ultimate customer.

This is not to say that there have not been efforts at creating more generic systems for seamless Information flow. Many great strides have been made in this area but, as we will see through the following discussion, no one has as yet sought to develop a model generic enough to be applicable to virtually any traditionally structured organization.

Before summing up this chapter, we will explore two more types of limited Information flow automation: Business-to-Business (B2B) and the Workflow Automation Initiative.

2.5.1 Business-to-Business Initiatives

The Electronic Data Interchange (EDI) frameworks such as UN/EDIFACT [ISO 9735 1991] and SWIFT allow one organization to connect to other organizations for services. However, such infrastructures have disadvantages for they allow business-to-business connections only [Hoffner 2000]. Additionally, they tackle the problem of Information flow using low-level message-type specifications, and are time-consuming and expensive as well.

B2B Frameworks Examples

- *An architecture for cross-organizational business processes* proposed an electronic connection between two business units to enable cross-organization business processes [Hoffner 2000]. However, this work focused on electronic Information exchange between businesses making use of each other's services. It does not dictate any norms or rules on structuring the Information flow within the organizations that are sharing services. Additionally, it does not address how the customers would interface with the organization for adding data to its internal Information flow.

- *An e-business integration and collaboration platform for B2B e-commerce* gives a framework to design a platform that supports collaborative business [Bhaskaran 2001]. It also proposes a model to assemble an e-business application for collaboration. However, although this model addresses B2B collaboration quite effectively, it still isn't a generic framework that any Web-based interaction like e-education, e-training, or e-governance can readily follow. It, too, lacks a model to interface with the customer as an extended part of the organization's internal entity.

- *An inter-college e-commerce system* proposed a framework by which colleges are able to extend courses to each other by seamlessly integrating legacy systems. The configuration also automates the flow of data between colleges and the Web. It is a significant improvement over the approaches discussed above as it provides a means of interfacing the customer, that is, the course-takers, with the college domain. However, since it achieves this by assuming the course-taker to be a part of the college domain and not an external entity, this framework is not generic enough to be applicable to a spectrum of interactions between organizations and the customer.

2.5.2 Workflow Automation Initiative

Another approach to automating the Information flow is Workflow Automation [Baggio 2003] [Aversano 2002]. Workflow can be generally defined as the flow or progress of work done by a company, industry, department, or person. Lately, Workflow has come to be associated with automation concerned with providing the specific information required to support each step of the business cycle [Georgakopoulos 1995]. We observe that Workflow solutions focus more on 'configuring and sequencing' the Information flow to suit a particular kind of business, rather than on providing a generic and intuitive Information flow infrastructure. While the results of Workflow Automation are of value, they do not provide an overall foundation of seamless automated Information flow. We conjecture that a Workflow engine would be greatly enhanced when implemented on top of an automated Information flow system. The Workflow engine is then not required to search for Information, but only to associate the Information with tasks that populate the organization's business process. In other words, the automated Information flow will form an infrastructure that enables a powerful Workflow tool, but the Workflow engine is not a means to achieve Information flow automation.

2.6 Summary of the Survey

It is evident from this brief survey of pre- and post-Internet era automation efforts that the technical challenges of the 1980s through the mid 1990s were different than they are in the 21st century, and so, obviously, were the solution approaches at that time. The key differences between then and now are the innovations that eliminate the need for paper documents at any point in the business process. Current technology can ensure the authenticity and security of digitally created "documents." Service providers now have the technological support to create an electronic form that can be completed remotely by the customer. The proper access and digital signature ensure the authenticity of the electronic document, which can be treated just as a paper document. This type of infrastructure was a mere dream not more than a decade or two ago.

In today's world, even the simplest of e-commerce Websites are more advanced than what researchers in the 1980s would have predicted. Today, on the Internet, we can search for a product catalog, select and pay for it, and receive it via mail on our doorsteps without ever touching a piece of paper or pen. So, we do witness paperless commerce in our everyday life, which is testament that complete automation is possible using current technology. The problem is not whether complete automation is now possible or not, but how to create the infrastructure for the Information flow that will automate even the most complex, largest, and oldest of organizations.

Chapter 3 – Conceptual Components of SIF

3 Conceptual Components of SIF

This chapter introduces the conceptual components of SIF. Through describing the elements that comprise SIF, we will illustrate the framework of this method and explain the functionality and universality, which make it an ideal solution for introducing Structured Information Flow within an organization. The goal of SIF is to provide existing, mature organizations with a framework to implement complete Information flow automation and seamless integration with external organization entities, as described in Chapter 1. This goal is wholly contingent upon the efficient and proper handling of Information, which is entirely contained and conveyed electronically. Thus, as the conceptual components will show, *Information* and its handling are central to the purpose and process. Likewise, the conceptual components of SIF all deal with the flow of Information. To facilitate the understanding of how the components of SIF apply to an organization, we must understand the structure of a typical organization targeted for SIF. Therefore, the description of the conceptual components will be prefaced by an explanation of organizational structure. This foundation will provide a context for examples, which will illustrate the concepts of SIF.

3.1 Components of Organization:

Typically, an organization is composed of three things - People, Information, and Work Structure. **People** usually perform and manage work collectively to achieve the objectives of the organization. **Information** is invariably created and updated in the process of performing and managing work. **Work Structure** is comprised of the ground rules by which people perform their work. For example, a manager is responsible for engaging his group of people to achieve a certain task, and each group member will report to their manager any difficulties in performing their task.

In Figure 3.1, we present a simple case of two people performing their task and, in the process, producing and updating Information. The work structure in this case is simply a sequence; that is, person B can perform a task only after person A has initiated the task.

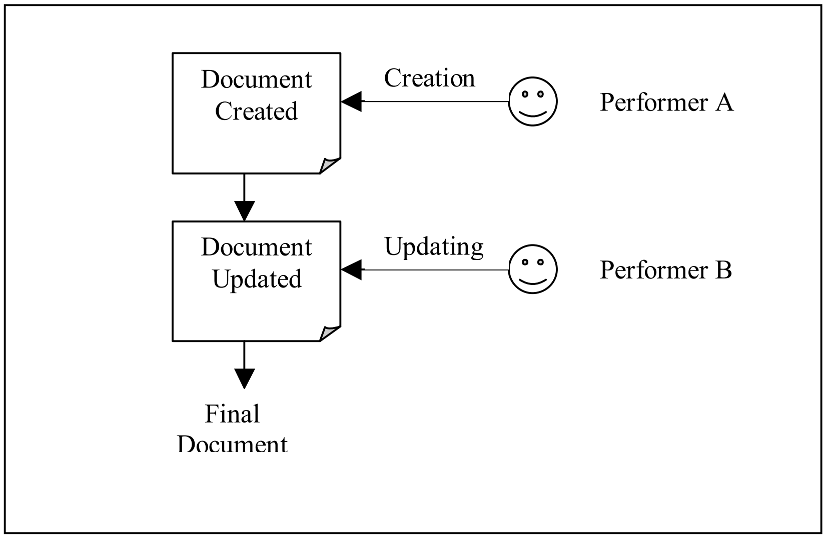


Figure 3.1: Performers Creating and Updating Information in an Organization

The basic organizational structure is fairly simple. But, given that this proposal of SIF is as a universal solution to the task of automating the flow of Information, it will be beneficial to examine an organization from an Information perspective. From an Information-centric view, we again observe that an organization is composed of three main elements: Information, a Work Structure of organizational entities that process the Information, and the Rules for Information Flow.

<u>Information</u>

It is easy to understand that, while performing tasks in any type of organization, Information is created for various purposes such as tracking progress, quality assurance, expense reporting, etc. The timely distribution and collection of correct Information is absolutely necessary for proper functioning of the organization.

<u>Work Structure</u>

The Work Structure is basically the formal relationship between the people working in the organization. For example, in Figure 3.2, A is superior to B, C, and D. We refer to the work structure of an organization as the **Topology** of the organization. Relative standing of a worker in this topology is generally reflective of the rights that a worker enjoys, including certain rights to Information. If the structure is not clearly defined, with appropriate rights allotted to each worker, obviously there will be lapses in communication that could negatively impact productivity and efficiency.

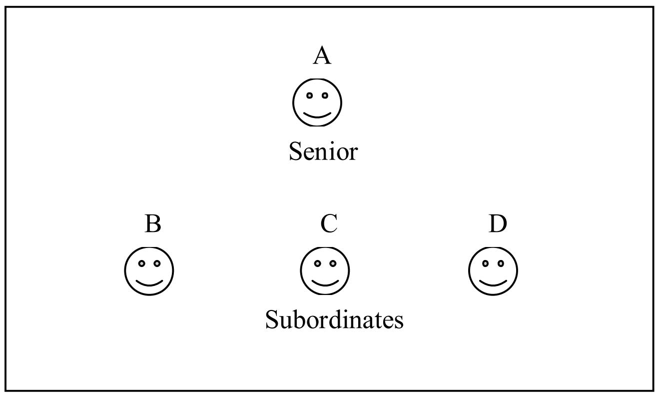


Figure 3.2: Work Structure of a Hypothetical Organization

<u>Rules for Information Flow:</u>

The Rules for Information Flow are the logic that governs when the Information should flow and to whom. For example, there are certain factors like timing/scheduling of tasks that trigger the flow of Information to a processing entity that has the right to process the Information piece. The lack of an automated system for triggering the Information flow is an invitation for a breakdown in processing.

From either an organizational or an Informational perspective, we can see that, in general, organizations share a basic structure that is conducive to the Information flow improvements offered through SIF.

Just how are those improvements effected? The conceptual components of SIF are designed to accommodate and reflect the above three basic components of an organization. SIF provides for the timely distribution and collection of correct

Information, a clearly defined work structure, and an automated system for triggering Information flow. This similarity between the components of SIF and the organizational structure satisfies the requirement of Issue 2, mentioned in Chapter 1: Reflecting the Organization's Structure. The following sections will define the conceptual components of SIF and explain how they meet each of the other requirements identified in Chapter 1. We have already identified three components through this discussion of organizational structure: **Topology**, **Information**, and **Rules**. Additionally, there is one more conceptual component of SIF, called the **Interface**, which is a mechanism to implement the rules that govern the Information flow. This additional component of SIF is designed to resolve Issue 1 – Intelligent Routing.

3.2 SIF Components

The conceptual components of SIF are:

1) Topology – The conceptual view of the entities in hierarchical formation.
2) Information – The business data that is processed by the entities.
3) Rules – The constraints and privileges to the Information that structures its flow.
4) Interfaces – The mechanism to implement the rule based Information flow.

To fully explicate the process of SIF, the significance of each of the above SIF components is illustrated in the remainder of this chapter using real life examples.

<u>The Organization Used for Examples</u>

A Police Department, as an organization familiar to the general public and representative of large government entities, will serve well to illustrate the components of SIF. This organization is also exceptionally well suited for explaining SIF because:

- It is a large department consisting of various sub-departments like traffic, narcotics, etc., which allow us to demonstrate the power of abstracting various functional units in SIF.

- Information plays a very critical role in the functioning of this department. Additionally, it relies on the input of sensitive and often anonymous Information, such as a report of suspicious activity in the neighborhood. Such diverse Information needs of the department help us demonstrate the versatility of SIF for handling a variety of Information, ranging from sensitive to shareable.
- Information flow is mission critical for this organization. Correct Information at the right place at the right time holds the key for operations of this organization, and the ability to accommodate this in the infrastructure that SIF provides will speak volumes for this approach.

The following sections describe the SIF components in detail.

3.2.1 Topology

SIF utilizes the hierarchical work structure of the organization to build its conceptual nodal topology. nodes are the conceptual representations of the roles in the organization that perform some work and process the Information in doing so. In short, nodes can be deemed as Information processing points.

Figure 3.3 shows the simple SIF topology skeleton for the hypothetical organization shown in Figure 3.2. Please refer to Table B.1 in Appendix B for frequently used symbols and their meaning.

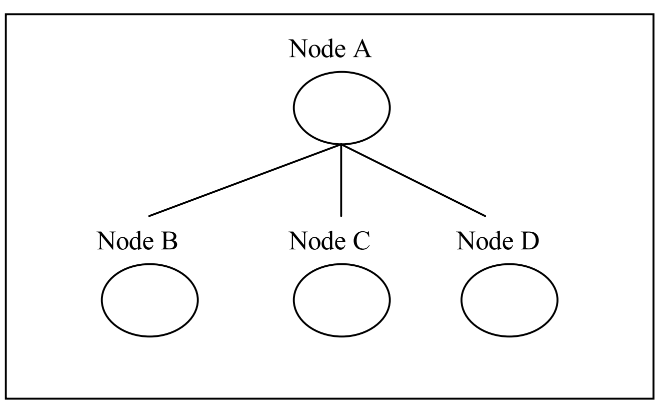


Figure 3.3: SIF Topology Skeleton for the Hypothetical Work Structure

Why does SIF use hierarchical topology only? This is because the hierarchical relationship between the nodes provides a very intuitive way of structuring the access and flow of Information by and between the nodes. This structure, which can be imposed on the Information access by virtue of hierarchical topology, is easily illustrated by a simple example: If, in the existing structure of the organization, A is superior to B (A occupies a higher level in the topology relative to B along the direct chain or hierarchy), A can access B's Information for viewing, while B cannot access A's Information.

In SIF, hierarchical arrangement of the nodes is the source of Information access rights that are granted to the node. More specifically, the relative hierarchical standing of the node in the topology determines certain Information access rights that a node can exercise. It is not difficult to observe that most organizations with which we interact, like banks, commercial institutions, educational institutions, etc., can easily be mapped to a hierarchical work structure. In fact, the most natural work arrangement of an organization is captured by a hierarchical topology. However, there are certain scenarios in which the hierarchical relationship between certain nodes is not obvious. The Information Based Precedence Table, described in Appendix A, is a method proposed in this thesis that allows us to establish a clear hierarchy between nodes when one may not naturally exist.

3.2.2 Information

Information is that human perceptible data that is created to perform, track, validate, etc., the functions for achieving the organization's aim [Boisot 2004]. The organization may be of any type - product-based, service-based, small, medium, or large - it will invariably need to create and maintain Information for a plethora of reasons. For purposes of this paper, the term **Information** actually refers to several types of Information: Captured, Created, Restricted, and Universal, and some include sub-categories. Each of these terms will be defined separately in the following paragraphs, as well as two methods for creating a pragmatic framework for Information flow. As discussed earlier, Information and Information handling are at the core of organizational structure. Thus, Information is central to the concepts of SIF and is defined as follows.

3.2.2.1 Information Generation

Information is generated in SIF two ways: it is Captured from the external domain, or it is Created in the domain internally. Captured Information and Created Information are explained below.

3.2.2.1.1 Captured Information

When an entity, external to the organization, registers the Information with the organization, it is termed ‘capturing’ of the Information. As depicted in Figure 3.4, the example of the Police Department will illustrate how this activity may occur in a real world scenario: A person notices some suspicious activity in the neighborhood and wants to register a concern with the police department. In the SIF-based model of the police system, the person (external entity) can use Internet access to reach the appropriate police department node, such as the area inspector node (node B). There he can access the appropriate form and populate it to register the Information. The instant the Information is submitted, it’s considered registered with the police department, and from that point on it becomes a document in the police organization domain. The node where the Information was registered (area inspector – node B) becomes the owner of the registered Information, and the Information physically resides at that node.

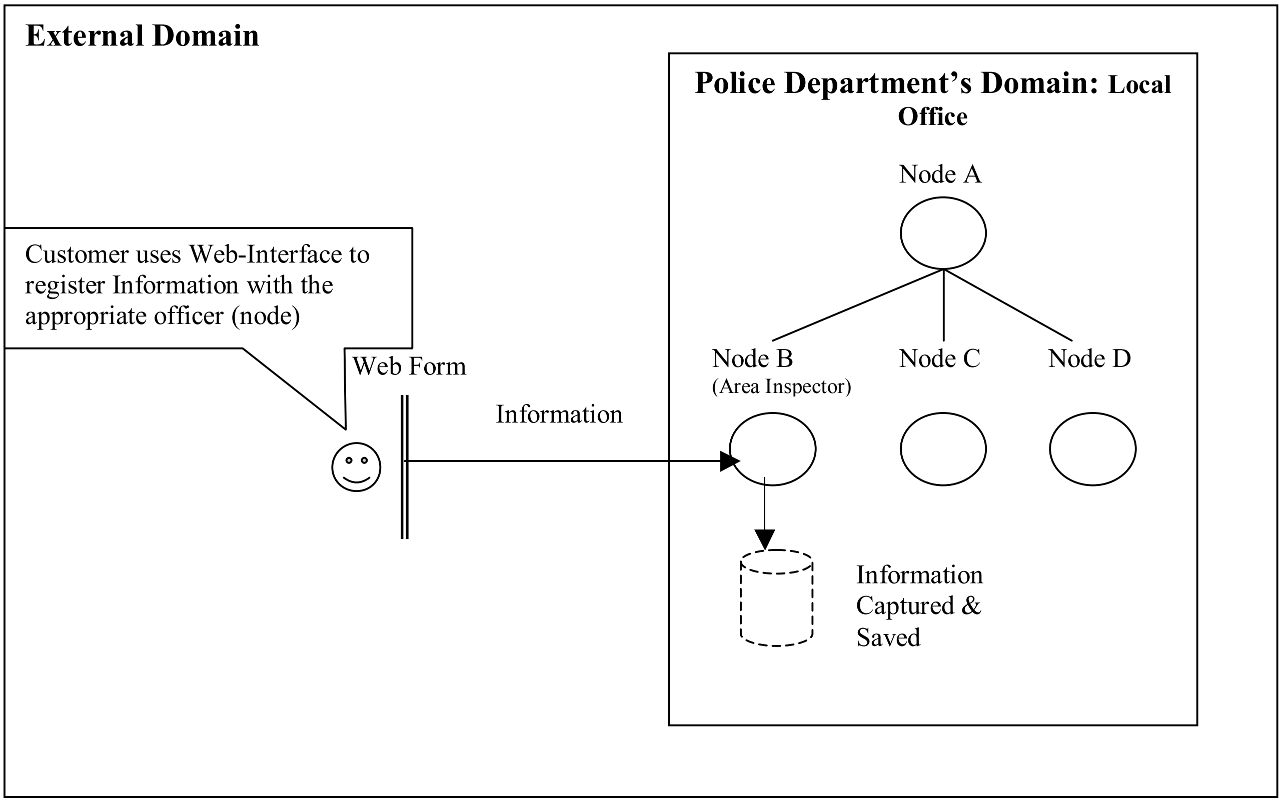


Figure 3.4: Capturing Information

3.2.2.1.2 Created Information

When an Information piece is generated within the organization rather than captured from outside the domain, it is termed 'creating' the Information. For example, if any particular department, or node, within the police organization generates a report, perhaps on the frequency of suspicious activity at a certain address, the report is Created Information. The node that creates the Information is the owner of the Information, and the Information resides in the physical databank of the node.

3.2.2.2 Information Categories

Just like in real life organizations, Information existing in the SIF system is characterized as 'universal' (Unrestricted access), universally accessible without requiring any permission; and 'restricted,' Information to which the access needs to be controlled and managed based on rules and rights. SIF manages these types of Information as follows.

3.2.2.2.1 Restricted Information

The Information in the organization meant to be viewed and used only by certain roles is termed 'restricted' Information. Circulation of and access to such Information is controlled and limited only to the nodes that have rights and privileges to the restricted Information piece. The first owner or the owner of the Information determines if the Information is restricted and labels it as such at the end of the *capturing process* or *creating process*. If labeled as restricted, it can be accessed only by the nodes with rights to access it, either hierarchically or because the owner has granted access rights.

<u>Restricted Information with External Designator</u>

In some instances, Captured (registered from the external domain) Information pieces require some kind of receipt sent to the external entity that registered the Information. For example, after registering the suspicious activity in the neighborhood, a concerned citizen would like to receive a receipt indicating that the Information has been captured and received by the intended node. This receipt can be sent to the external domain entity as an email reply, or retrieved via password access, or a variety of other ways. The receipt, which is restricted for retrieval only by the designated external entity, is termed Restricted Information with External Designator.

3.2.2.2.2 Universal Information

Information that can be viewed by any node as well as any external domain entity is called Universal Information. The name 'Universal' itself indicates that it is universally accessible - but *not* universally changeable. A police department press release is a good example of Universal Information; everyone can read it but no one can change it except the owner.

3.2.2.3 Pragmatic Considerations

As described in Chapter 1, mature organizations with cumulative legacy systems are main targets to be benefited by SIF. One can expect to be dealing with an enormous amount of data that is captured, created, and processed by such organizations. Issue 3 in Chapter 1 dictates that a standard framework of Information flow for such organizations should feature a pragmatic approach built into the framework itself. SIF deals with this issue by inculcating two concepts in SIF, namely: Information Aggregation, which utilizes principles of abstraction to pragmatically consolidate restricted Information at nodes for easy retrieval; and Information Pooling, which is a logical view of a collective set of Universal Information that facilitates its dissemination.

These two concepts are discussed below.

3.2.2.3.1 Information Aggregation

It is commonly observed that in day-to-day functioning at the office, we need to access Information at times but we do not know where it resides. Sometimes the Information is easy to find, sometimes the search becomes a laborious process. As we already know, in SIF, the Information owned by a node physically resides with that node. As shown in Figure 3.5, node A's physical data bank contains the information pieces owned by K and L.

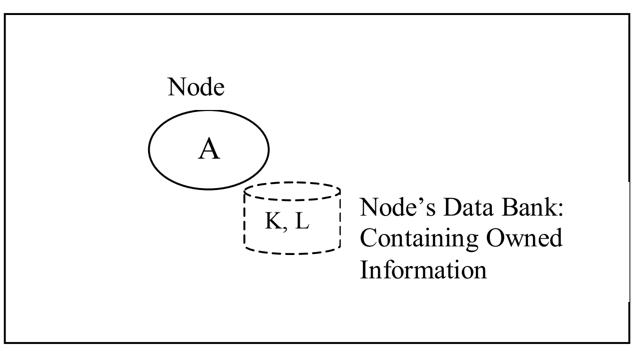


Figure 3.5: node and its Physical Data Bank

This concept can be explained further by example: a node X in the sub-department of Criminal Justice needs to access the Information residing in the Narcotics sub-department. The officer at node X does not know precisely which node in the Narcotics branch owns that Information. Typically, the officer would need to invest a fair amount of time in sending some emails, placing some phone calls, or submitting requests that would be routed to the appropriate officer in the Narcotics branch, who would then decide if that Information can be shared or not. If the Information can even be shared, the act of sharing itself will consume some time. These are the kind of inefficiencies that have come to be accepted in today's workplace. With SIF-based Information flow automation, however, there is a single point in a sub-organization where all the Information owned by the sub-organization is aggregated. SIF provides an elegant solution for Information Aggregation, which is continuous and self-sustaining, as well.

3.2.2.3.1.1 Aggregates in SIF

The Logical Information Bank, or the Aggregate, as depicted in the Figure 3.6, is the logical view of all the Information residing at the node. An Aggregate in SIF is the consolidated logical view of the Information owned by the particular node and its immediate and direct[1] subordinates. Hence, the Aggregate at the node at the lowest level contains just the Information owned by the node itself. However, the Aggregates at the nodes at higher levels have a consolidated view of all the information owned by the immediate and direct subordinates at each subordinate level.

[1] Immediately superior or subordinate node is the one that is right above or below a node, respectively, while a directly superior or subordinate node is the one that is in the path above or below the node, respectively.

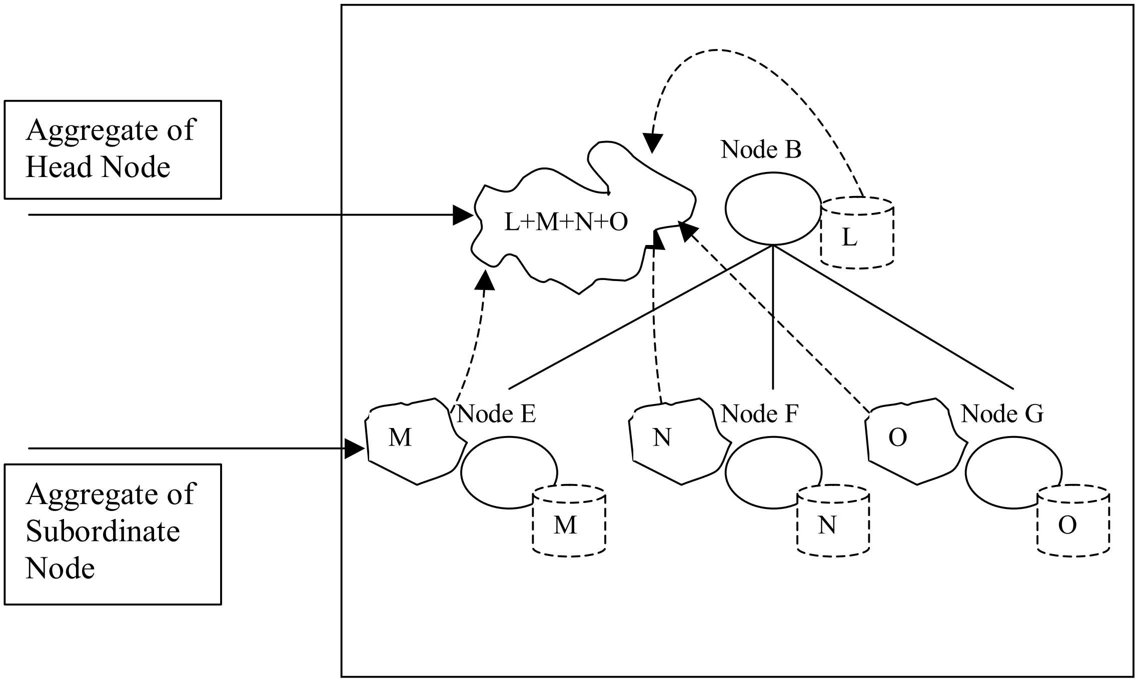


Figure 3.6: Information Aggregation

Practically, what this means is that the cumulative Aggregate exists at a head node. In other words, a head node's Aggregate is the logical consolidation of all the Information existing in the sub-organization, thus becoming the single point of reference to all the Information owned by the sub-organization, just like node A in Figure 3.7. Any node from outside that sub-organization searching for Information in that particular sub-organization need not go from to node to node, but can simply search the head node's logical data bank.

Thus, the scenario with node X in the sub-department of Criminal Justice is greatly simplified. In a SIF-based system, node X will simply go to a single point, the head node in the Narcotics sub-department, to view all the Information titles and choose the Information of interest, rather than going to multiple points to search for the Information.

Whenever any Information is created or changed or deleted, the Information Aggregates existing at all the appropriate nodes in the tree must be updated. Note that in Figure 3.7, the Aggregates of nodes B, C and D are being consolidated into Head-node A's Aggregate. Aggregates of nodes B and D comprise the information consolidated from their respective subordinate nodes. node C does not have any subordinate nodes.

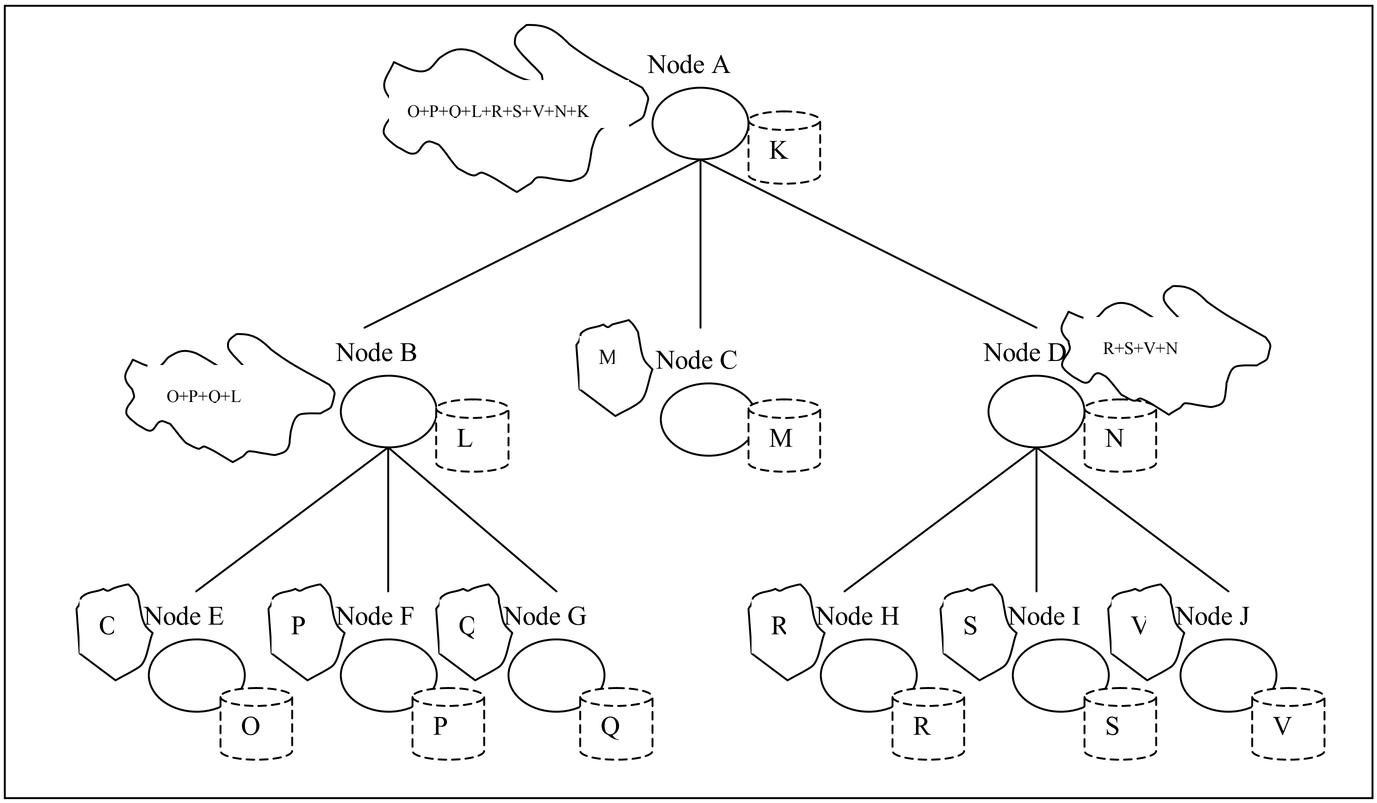


Figure 3.7: Head node as the Information Aggregation Point of the Department

3.2.2.3.2 Universal Information Pooler

As the name suggests, the Universal Information is supposed to be accessed by one and all. Hence, it must be freely and easily accessible, regardless of one's position in the Topology. For the sake of simplicity of search and access of Universal

Information, a logical set or pool of all the Universal Information in the organization is designed to reside at every node in the system.

3.2.3 The Rules

Every piece of Information produced in an organization has a target audience inside or outside the organization. For example, for the budget reports, middle and upper management roles are the audience; for an expense report of a business trip, the internal auditor may be the audience. Our observation is that, in most organizations, such Information pieces are delivered to the intended audience via hard copies or, in better cases, via emails. The Information pieces reach the audience, but not in the most efficient manner possible. As mentioned previously, even if there are IT systems involved for the Information delivery, in the vast majority of cases, it is not fully automated using intelligent systems. SIF provides a set of rules to enhance the efficiency of Information flow.

In SIF, the intelligent and automatic routing is made possible by the rules that govern the Information access and flow. The rules defined in SIF enable the system to make a real-time decision, when the request to access an Information piece is submitted, whether to grant that access or not. This, in turn, determines whether the subsequent Information flow takes place or not.

The rules in SIF are based on two factors derived from an interdependent combination of characteristics of two SIF components, nodes and Topology. These two factors are Hierarchical Standing of the node and Ownership Rights of the node. Together, they define the rules that govern the Information access and flow.

3.2.3.1 Default Rules

The basic rules are kept to a minimum and are relatively simple. The first rule is a default condition wherein certain kinds of access to a piece of Information owned by node Y can be granted to a node X based on the fact that node X is senior to node Y. So the default rule states:

A node can view any Information owned by any of his subordinates in the direct chain of the hierarchy.

This rule follows from the fact that, in most hierarchical organizations, roles report to one of their direct superiors. Hence, in SIF, the access to view Information is granted based on the superior hierarchical standing of nodes, as the superiors need to constantly monitor the work of subordinates for general feedback and guidance. However, SIF understands that, while monitoring the Information, superior nodes should not have free reign to edit or delete subordinates' Information. Hence, edit or delete accesses to the Information are not granted using hierarchical standing or the default rules.

So, how does the system know whether to grant edit or delete access to a node? This is determined by the next rule discussed below.

3.2.3.2 Customizable Rules

When a node captures or creates Information and becomes its owner or the first owner, along with the ownerrship comes a full set of rights over that Information, namely: View, Edit, and Delete rights. Although senior nodes can view the Information, only the owner can edit or delete the Information, or extend those rights to another node. SIF incorporates two rules governing the extension of rights.

3.2.3.2.1 Extension Rule 1

An owner node can extend its right to edit or delete Information to another node.

However, only one node can be editing or deleting an Information piece at a time, taking the Information integrity issues and technical constraints into consideration. It logically follows that when an owner node extends edit and delete rights to another node*, it releases its own rights for the same*. In fact, the original owner node (that extended the rights) loses its ownership, and the node that received the edit and delete rights now becomes the new owner of the Information. Therefore, this rule is qualified by the condition: *an Information piece can have only one owner at a time that can exercise edit and delete rights*.

3.2.3.2.2 Extension Rule 2

A node can extend view rights over its Information or that of any direct subordinate's to any other node.

We know that a superior node can view a subordinate's Information at any time. But there are situations when a subordinate may need to access some of the Information owned by a superior's node. In that case, the subordinate may request view access to the Information from the superior and the superior may choose to extend the view right to the requesting node. In case the superior node does not take an action to that effect, the subordinate may choose to request the superior's superior to provide view access to the subordinate's Information. A relatively super-superior node can extend the view right on its subordinate's Information to a super-subordinate node, as depicted in Figure 3.8.

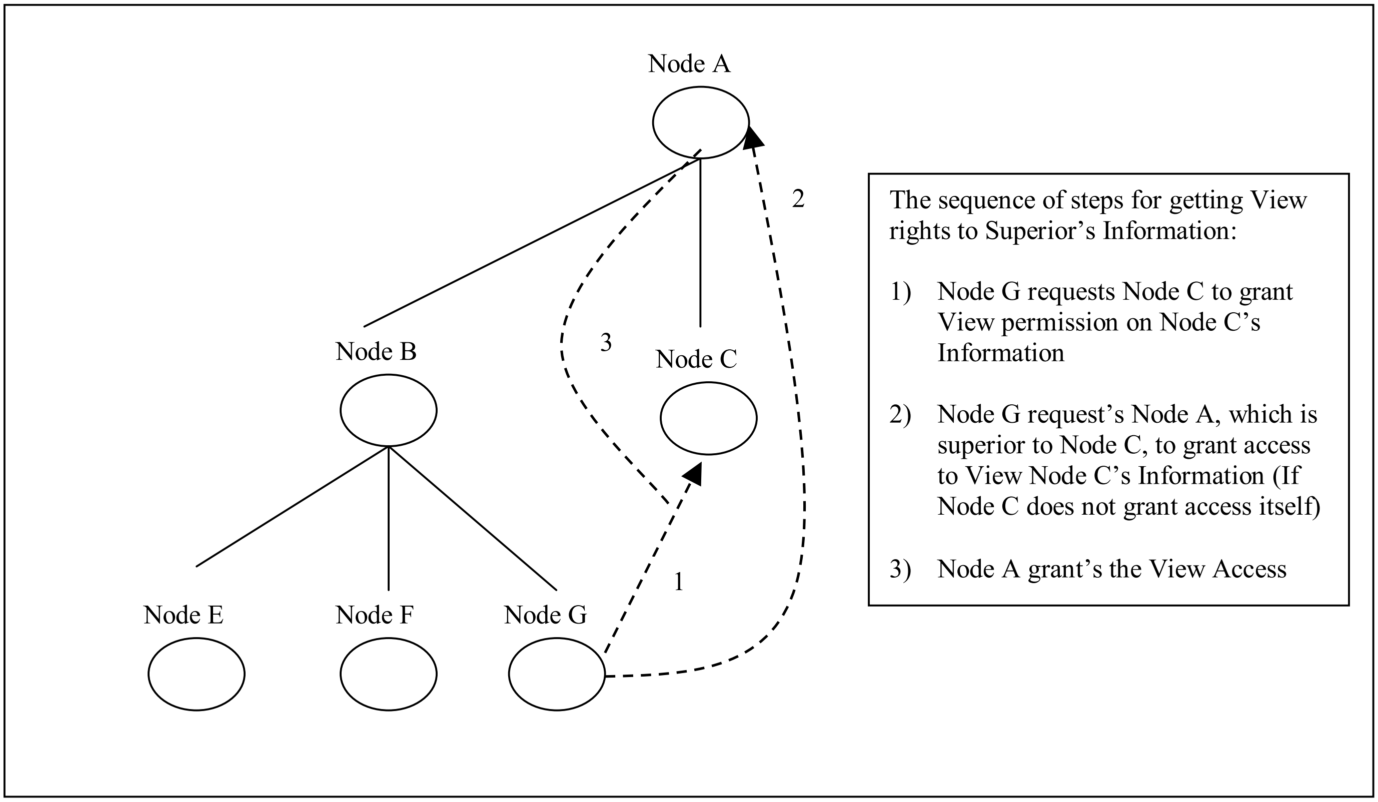


Figure 3.8: Getting View Right on Superior node's Information

3.2.3.3 The Nature of Default and Customizable Rules

While introducing the rules, we called the rules based on the hierarchical standing of the node the *default rules*, and the rules based on the ownership rights the *customizable rules*. The difference between these two types of rules is straightforward. The rules based on hierarchical standing of the node, *default rules*, are general in nature and will remain the same for any organization. To wit, in any organization where A node is superior to B node, A will exercise certain rights based on the rules that are organization-independent. This addresses Issue 3 of Chapter 1 – Organization and Technology Independence. These rules are in effect by default as soon as the hierarchical skeleton of the organization is formed in SIF. However, the rules based on the ownership are specific to the functioning of the organization. Two organizations may have the same hierarchical skeleton but may have a totally different pattern of Information ownership and extension of rights and, hence, a different set of Information flow paths. The ownership rules and the extension of rights customize the Information flow path specific to the needs of an organization.

This gives rise to the question of where the Information about the various ownership and/or extended rights is stored. The answer is discussed below.

3.2.3.4 Core Information and Information Attributes

Any Information regarding ownership of and rights to the Information is stored in the Information Attributes. In SIF, the Information Attributes feature is one of two components composing Information, as defined below:

- **The Core Information** –Information that needs to flow around for use in the organization, such as a report.
- **The Information Attributes** - The component of the Information that contains Information about who has what rights over the core Information, who created the core Information and when, who is the current owner of the core Information, etc.

Information Attributes “piggyback” on the Core Information that flows from one destination to another and enable the system to decide whether a particular node can view the Information only, or edit/delete it as well.

The Information Attributes contain the data about the Information that is required to make a decision about whether the requestor node should be granted access or not, such as the hierarchical standing of the owner node, for example. Figure 3.9 shows the Core Information along with the Information Attributes required for granting access to the deserving requestor nodes.

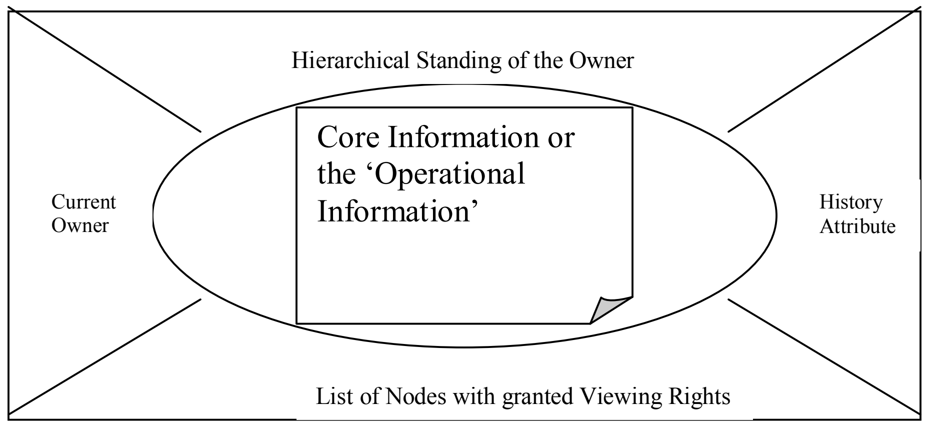


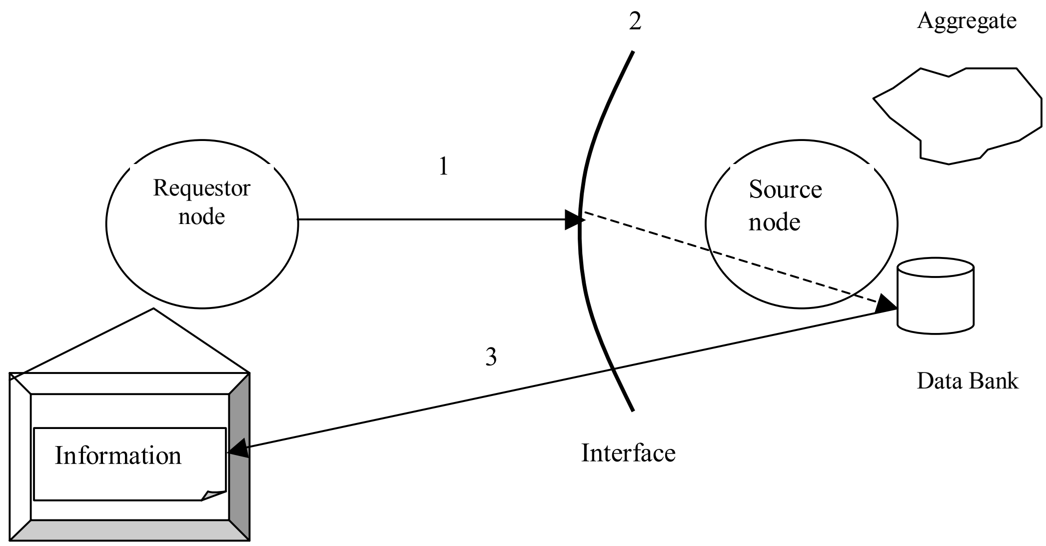
Figure 3.9: Core Information and Information Attributes

3.2.4 The Interfaces

The final component of the SIF framework, the Interfaces, performs the function of decision-making to grant or deny the access, as discussed in the following section.

3.2.4.1 Information Interfaces in SIF

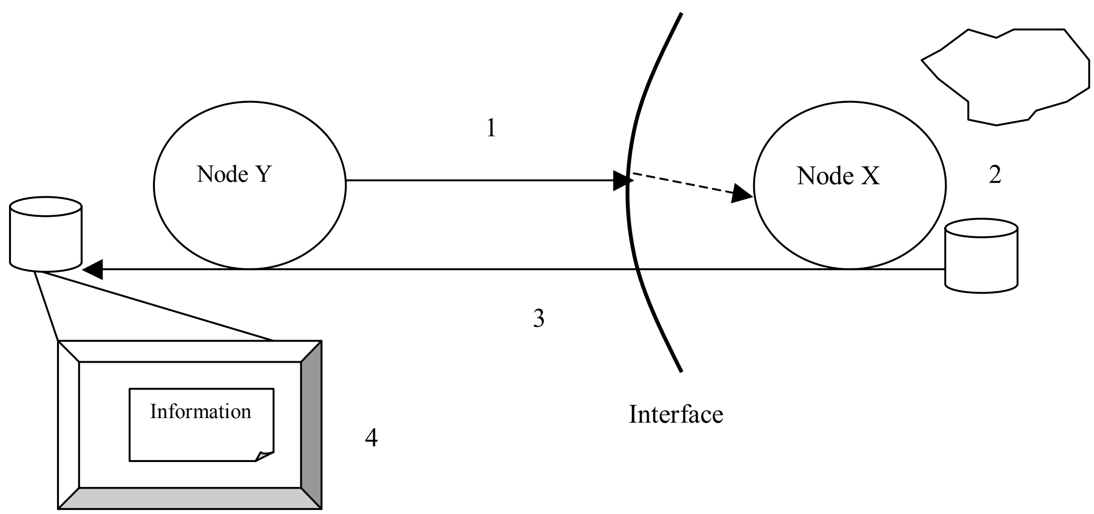
Interfaces are the conceptual component of SIF that implement the logic for the rules-based Information flow. As discussed in the last section, the critical Information required for the decision-making – whether access is granted to a particular node or not – is stored in the Information Attributes. This Information must be analyzed based upon the pre-defined Information flow rules at the time the request for Information access is made. Interface is the layer that contains this logic. Every node is logically surrounded by the Interface. Every request for Information access made by any requestor node must pass through the Interface of the grantor node. If the grantor's Interface decides that the requestor node can view the requested Information, it displays the Information for viewing in the console window of the requestor node, using appropriate software, as shown in Figure 3.10. In this example, the Information was accessed for viewing but no physical Information flow occurred. Information flow takes place in the scenario described in the next paragraph.



1. Information Request is placed at the Interface.
2. Interface decides whether the Information request should be granted or not.
3. Interface grants the access and the Information is viewed at the requestor node's computer.

Figure 3.10: Information Access and Display Via Interface

In this circumstance, Y requires edit/delete rights, which only an owner can exercise, on an Information piece owned by X. If X agrees to grant ownership rights to Y, it simply changes the attribute of its Information piece to designate Y as the new owner. The moment Y becomes the new owner of the Information piece, the Information Piece flows from X's physical databank to Y's physical databank, as shown in Figure 3.11 along with other prior steps. This is the Information Flow. Corresponding to the changes made in X's and Y's databanks, appropriate changes are made to all the Information Aggregates in the sub-organization.



1. Request to grant Ownership right.
2. Ownership right granted to Y by X.
3. Interface allows the Information to flow (physically move) from X's physical databank to Y's databank.
4. Y can View, Edit or Delete the Information piece on its computer.

Figure 3.11: Information Flow Via Interface

At any point in time, when a change in the Information bank takes place, the change needs to propagate throughout the entire model to be reflected in the Information Aggregates, and so on.

The next chapter provides a more formal approach, using State Models and State Transition Rules, to describe when and how changes in the SIF model take place when triggered by various events in the system.

3.3 Summary

This is the fundamental chapter of the thesis, which introduces the conceptual components and key concepts of SIF. The chapter begins by enumerating the components of SIF – Topology, Information, Rules, and Interfaces – with relevance to real life organizations. All the components and related concepts are discussed in detail, though informally. In this chapter, we introduced a police department as a real-life organization appropriate to explain and test the SIF concepts for all the subsequent chapters. For the purpose of explaining the model, multiple symbols are defined in this chapter, which in subsequent chapters will be used for the formal visual representation scheme.

Chapter 4 – Formal Description of the SIF Framework

4 Formal Description of the SIF Framework

The theory of Information flow and the conceptual components of the solutions offered through SIF provide a view of the ideal office, effortlessly operating in a paperless environment. Although, at present, this ideal may seem difficult to achieve for most large organizations, it is actually an attainable goal. As discussed earlier, this goal is only possible with the advent of a standard framework to structure the Information flow automatically. Pursuing this goal requires a methodical strategy, a plan of attack. In this chapter, we present the formal definition of the functions of SIF, using notational language and State Models to formalize the components introduced in Chapter 3. This chapter, then, becomes a map to this entirely reachable goal.

As the conceptual chapter illustrated, SIF enables existing, mature organizations to create a paperless office through automation and seamless integration of Information flow. The technical exposition that follows describes the proposed structuring of the Information flow within the organization, as well as the data flow into and out of an organization. The notational language and State Models will illustrate how SIF discriminates, based upon the Information type and the destination type, to determine valid Information flow paths.

4.1 Theoretical Model Components

In Chapter 3, the conceptual components of SIF were presented. The collective definition of these components – Topology, Information, Rules, and Interface – for an organization at a point in time ‘t’ represents the snapshot of the SIF Information flow infrastructure of the organization at that time. This snapshot changes with every change in the conceptual components, for example, when Information is created or destroyed, or when the topology is changed.

The SIF conceptual components and changes in the components can be modeled theoretically using the notion of State Machine. Generally speaking, a State Machine is any device that stores the status of something at a given time, and can respond to stimuli to change the status and/or cause an action or output to take place for any given change [http://whatis.techtarget.com 2005]. The SIF model is such a theoretical abstract State Machine that can be defined using State and Transition Events.

- **State** – the snapshot of the SIF Model at time 't'
- **Transition Event** – the input or the change that causes the State of the Model to change

4.2 SIF Abstract State Machine:

Below is the formal definition of the SIF Theoretical Model or the SIF Abstract Machine.

4.2.1 SIF States

A State stores information about the past; that is, it reflects the input changes from the system start to the present moment [http://en.wikipedia.org 2005]. At the very start, the system is defined using various components that can change with time by externally or internally generated inputs. Any change in these components of the system causes the State of the system to change to a new State.

A SIF Model can be defined using the four components of SIF – Toplogy, Information, Rules, and Interfaces. Hence, the State at time 't' becomes a function of these four components at any time 't,' which can notationally be represented as:

$$S_t = (T, \Psi, \vartheta, \tau) \text{ at time t}$$

where:

- **T** is the Topology at time t which represents the Nodal Structure based on the reporting structure of the organization
- Ψ is the Information, Restricted and Universal, in the organization residing in the Physical banks and Logical Aggregates at time t
- ϑ is the set of Default and Configurable rules that constrain the Information access at time t
- τ represents the set of Interfaces at time t that implement the rules-based Information access

Figure 4.1a represents a simple State model at time 't' for a hypothetical organization. Please refer to Table B.1 in Appendix B for symbols and their meaning.

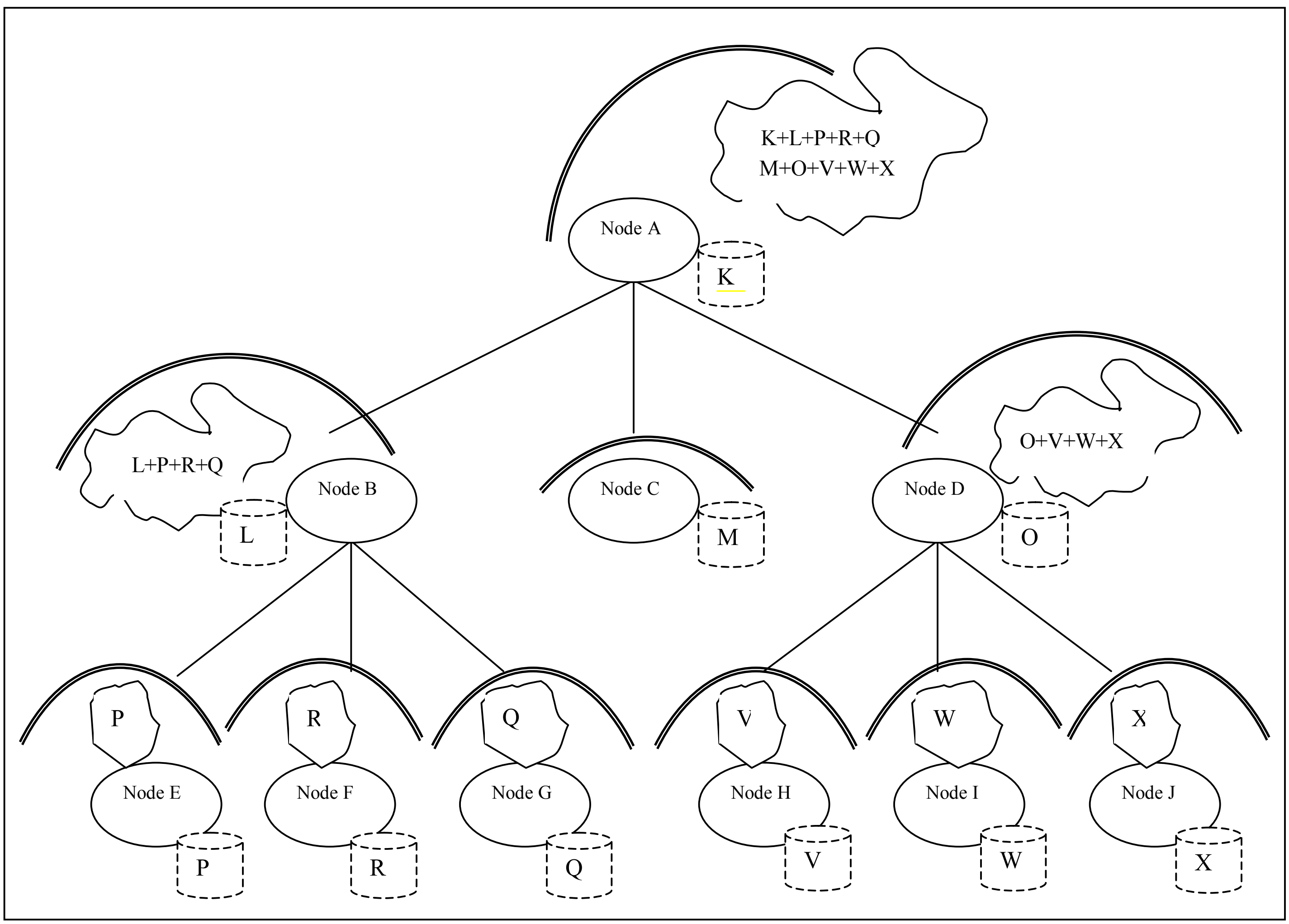


Figure 4.1a: The State of SIF Model for a Hypothetical Organization at Time t

For the sake of explanation, let's say node J is removed from the organization's reporting structure due to reorganization. This causes a change in the topology (T) at time 't', and the State transitions to the next State at time 't+1' to reflect this update in the model. Note that not only has the node J and its Physical Information Bank and Interface been removed from the model, but the Information piece X, which is now removed along with the node J, has been removed from the Logical Aggregates of its immediate and direct senior nodes.

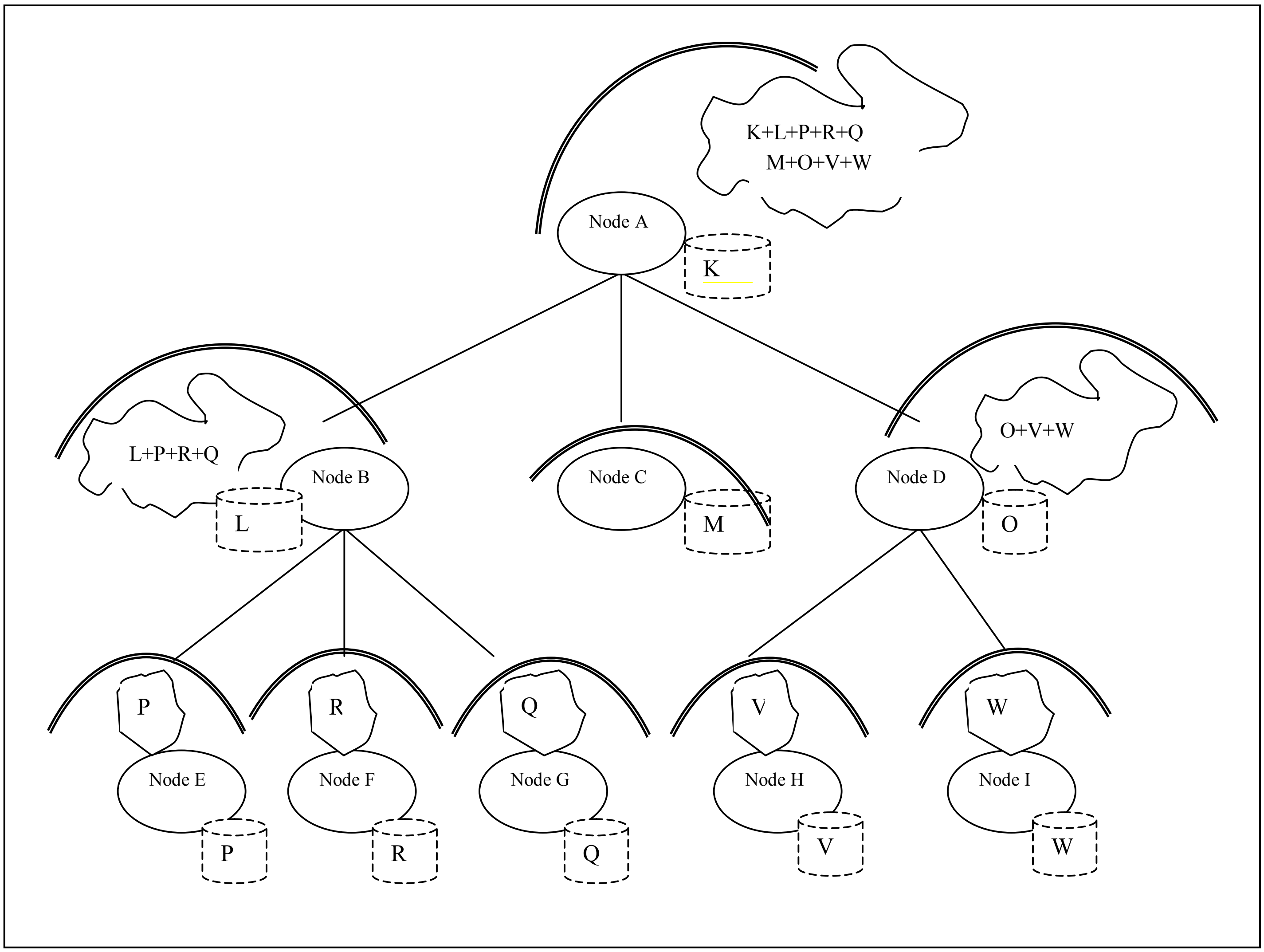


Figure 4.1b: The State of SIF Model for a Hypothetical Organization at Time t

4.2.2 SIF Transition Events

Transition Events actually capture two State Machine concepts of ‘Transition’ and ‘Event.’ These are not stand-alone concepts and can best be understood when considered together.

A Transition Event is any change in the SIF Model caused by the action in the organization. For example, in the above scenario, the reorganization is the action in the organization, removing the node J from the Model, which is the Event that caused the change in State. The change in the State is called the State Transition.

The Notational Language is defined to capture the States and Events at different levels of details.

At the highest level, Transitions and Events can be represented using the notational language as:

$$\delta_x = (S_t, E) \rightarrow S_{t+1}$$

where:

- δ_x is the Transition caused by the Event in state component X. For example, a transition caused by the change in the topology (T) will be δ_T
- S_t and S_{t+1} are the States at time ‘t‘ and ‘t+1,’ respectively
- E is the event that caused the change in any state component.

For a more granular level of representation, X and E can be replaced by more event-specific symbols. For example, adding node ‘N,’ which is a Topology (T)-based event will be represented by $\delta_T = (S_t, N_\pi) \rightarrow S_{t+1}$ where the new symbol N represents the specific instance of Topology component (node) that changed and π stands for the particular action of Addition (α) or Deletion (β) of node. In short, $\pi \equiv \alpha / \beta$

However, not all four components of SIF can change once the system is defined. Therefore, not all SIF components have associated Events that can cause State Transitions. The next section examines those components of SIF that do not change and why.

4.2.2.1 Changeable and Unchangable Components of SIF

Before addressing the unchanging components of SIF, it can be established as a given: *Information* is a constantly changing element, whether it is flowing within the organization or being introduced from external sources. Obviously, Information changes cause State transitions. Additionally, *Topology* is subject to changes reflecting any organizational shift. These changes also lead to State transitions.

Of the four components of SIF, *Interfaces* are the “brains’ of the system. SIF interfaces allow for an elegant execution of SIF rules. Interfaces create a logical layer between the node and the Information source that decides, based on the rights, whether to grant access or not. The rights, as explained earlier, are those circumstances surrounding a piece of Information that impact its flow, such as ownership and ownership, which are stored in Information Attributes. Interface just implements the access based on the rights. Hence, Interface does not change by itself and is **not** a source of State transition.

Within SIF, there are two types of rules: *Default* and *Customizable*. Default rules, based on hierarchical standing of the node, are general rules that stay the same for any organization. The Customizable rules, however, are based on ownership of the Information and lead to two types of changes: Propagation of Ownership, or Propagation of View rights. As explained in Section 3.2.3.4, various rights to the Information piece such as Ownership or View Rights, are stored in the Information Attributes. A change in the Customizable Rules is reflected in the change in the Information Attributes, which causes the State Transition. Thus, the Customizable Rules do not directly cause a State Transition, but the change in Information Attributes does.

The sources of State transition have been narrowed to only two components - Topology and Information. Further, the precise Events and the associated Transition Functions caused by the changes in the Topology or the Information can be named.

4.2.2.2 Topology-Based Events

Chapter 3 has already explained that Topology is based on the reporting structure of the organization. The hierarchical Topology of the SIF model changes with any change in the reporting structure of the organization. A change in the reporting structure can lead to three types of Topology-based Events: Addition, Deletion, and Rearrangement.

Any State Transition caused by a Topology-based Event is represented as:

$$\delta_T = (S_t, N\pi) \rightarrow S_{t+1}$$

Where $\pi \equiv \alpha / \beta$

Symbols are as defined in section 4.2.2.

- Addition of nodes – This type of Event occurs when new role/s is/are added in the organization, which is reflected as the addition of new node/s in the SIF Topology. New nodes begin with empty Information banks, which are then populated with the registered Information following the SIF rules for Information capturing and creation.

  The new node could be added either at the Leaf level or at any other level in the hierarchy. New nodes at Leaf level do not have an Aggregate and, so do not cause Information flow to populate the Aggregate. However, new nodes added at any other level in the hierarchy need to instantiate their Aggregate, which causes the Information Flow. The details of the State Transitions caused by Information-based events are captured in section 4.2.3.

Addition of a node J is captured in the notational language as:

$$\delta_T = (S_t, J\alpha) \rightarrow S_{t+1}$$

where Topology modifier π is replaced by symbol 'alpha' α, which indicates the action of adding a node, while J indicates the particular node to be added, the specific topology component that caused transition.

- Deletion of nodes – This type of Event occurs when existing role/s is/are removed from the organization, which is reflected as the deletion of node/s in the SIF Topology. Deletion of the node also leads to removal of the Information owned by the node, as well as the Aggregate from the model. If Information from the (to be deleted) node needs to be retained, then the ownership right of the Information pieces can be extended to other appropriate nodes. This function is addressed in the next section.

The event of Deletion of node J is captured in notational language as:

$$\delta_T = (S_t, J\beta) \rightarrow S_{t+1}$$

where symbol 'Beta' (β) indicates the act of Deleting the node J.

- Rearrangements of nodes – This type of Event occurs when existing role/s is/are rearranged in the organization, which is reflected as the rearrangement of node/s in the SIF Topology. Rearrangement of the nodes can be described in terms of Addition and Deletion, as discussed above. For example, a node J is to be replaced by a new node G. This could be broken down into two steps:
    a. Deleting node J
    b. Adding node G in place of J

Thus, as we see in this case, a single event can lead to a combination of other defined events. The above two steps could be represented using the notational language discussed under node Addition and Deletion bullets.

4.2.3 Information-Based Events:

Information is the most fluid of all the SIF components and is expected to change constantly as the organization goes through its daily functions. In section 3.2.3.4, we explained that Information has two sub-components: Core Information and the Information Attributes.

Any action in the organization that changes the Information can be translated as an Event in the model that changes either the Core Information, the Information Attributes, or both. Tracking the Core Information and Information Attribute events separately helps generate valuable statistics regarding the pattern and the nature of Information Flow in the organization. Typically, the Core Information is intended for use by the **people** in the organization, and the Information Attributes are mainly for **system** use only.

Below we enlist the Information-based Events in the SIF Model and describe the action in the organizations that can lead to such Events. We also discuss whether the Event is a Core Information and/or Information Attribute Event.

- Capturing of new Information – This Event occurs when Information is received from a domain external to the organization. The act of registering the Information in the system gives rise to this Event, which impacts both the Core Information and the Information Attributes. When new Information is created or captured, Core Information is produced along with Information Attributes.

Just as T indicates Topology, the symbol Ψ indicates Information. For Information-based events, the symbol E is replaced by the specific Information component that changed, represented by - KN_{ϕ} - where KN collectively represents that the Information piece 'K' belongs to node 'N'. ϕ represents the change in either the Core Information or the Attributes, as explained below. Other symbols are as defined in sections 4.2.1 and 4.2.2.

Capturing Core Information and Information Attribute Changes

In order to capture whether an action causes a change in Core Information and/or Information Attributes, we propose the following notation:

If symbol 'Lambda' λ signifies change in Core Information, and symbol 'Sigma' σ signifies change in Information Attributes:
$\phi \equiv \lambda / \sigma / \lambda\sigma / \sigma\lambda$ which means specific Information-based events could be of type Lambda or Sigma, OR both Lambda and Sigma where sequence does not matter.

An Information-based event affecting just Core Information for the Information piece K belonging to node N is represented in the notational language as:

$$\delta_{\psi} = (S_t, KN_{\lambda}) \rightarrow S_{t+1}$$

An Information-based event affecting just Information Attributes is represented in the notational language as:

$$\delta_{\psi} = (S_t, KN_{\sigma}) \rightarrow S_{t+1}$$

An Information-based event affecting both Core Information and Information Attributes is represented in the notational language as:

$$\delta\psi = (S_t, K_{N\sigma\lambda}) \rightarrow S_{t+1}$$

- Creating new Information – This Event occurs when Information is internally created. The act by an Owner of writing the new Information piece in the system gives rise to this Event, which Impacts both the Core Information and the Information Attributes. As mentioned earlier, when new Information is created or captured, both Core Information and Information Attributes are produced.

  We have already stated that any action causing an Event that changes both the Core Information and Information Attributes is represented as:

  $$\delta\psi = (S_t, K_{N\sigma\lambda}) \rightarrow S_{t+1}$$

  where K stands for the new Information piece that is created.

- Editing the Information – This Event occurs when existing Information is edited by the Owner or the Owner. The editing of the Information changes only the Core Information, and the Information Attributes as defined in Section 3.2.3.4 remain unaffected by this act.

  We have already stated that any action causing an Event that changes just the Core Information is represented as:

  $$\delta\psi = (S_t, K_{N\lambda}) \rightarrow S_{t+1}$$

  where K stands for the Information piece that is edited.

- Extension of Rights on the Information: This Event occurs when the rights to the existing Information are extended by the Owner or the Senior node to another qualified node. The extension of the Information Rights is captured in the Information Attributes, as defined in Section 3.2.3.4, and the Core Information remains unaffected by this act.

We have already stated that any action causing an Event that changes just the Information Attributes is represented as:

$$\delta\psi = (S_t, K_{N\sigma}) \rightarrow S_{t+1}$$

where F stands for the Information piece with the changed Attribute.

This section notated the exhaustive list of the Events and the underlying actions, along with the unique notational representation for State and State Transitions, caused by the Transition Event.

The notational language developed to represent the SIF theoretical model borrows the concepts below from the following sources:

- Finite triggers/events from FSM [Lee 1996]
- Infinite States from ISM [Bensalem 1998]

4.3 Testing the Notational Language and the Visual Representation

Let us now take a real-life example of an organization and model it using the theory developed in this chapter. The real-life example that we will model is a scenario from a typical police department using a hypothetical SIF-based system. The scenario describes an hour of a typical morning at an office of the police department. The scenario described below will be represented theoretically using the theory of States, Events, and Transitions captured in the Notational Language and the Visual Representation Scheme.

4.3.1 Scenario

It is 8 o’clock in the morning at the local crime-prevention police branch and there are two open cases. One case is assigned to Sub-Inspector ‘node A’ and another to Sub-Inspector ‘node B.’ The local office also includes the Head Inspector ‘node H’ and the new Sub-Inspector ‘node C,’ who reports to ‘node B.’

At 8:10 AM, Sub-Inspector A edits his case file to include the updates from the previous day’s investigation.

At 8:20, Sub-Inspector B receives new Information regarding his case from an external domain. At 8:30, B decides the new Information is worth urgent investigation. At 8:35, B extends the ownership right to C, who is the new Sub-Inspector. At 8:40, B also extends the view rights to A, who is the more experienced Sub-Inspector, to allow A to help C as needed.

4.3.2 Notational and Visual Representation

The following sections show the notational and visual representations of the various States as described in the above scenario. The actions described in the scenarios will lead to Transition Events for the States. In each of the sub-sections below, we will first describe the current State, then the Transition Event, which will be paralleled by the representation in the notational language. A visual description of the current State transitioning into the next State will follow.

In the visual representation, new symbols mean the following:

- Information about the case that A is handling = L_A
- Information about the case that B is handling = M_B
- Information owned by Head node H = O_H

More than one element of the model can be changed during one transition. The changed elements are identified in bold-letters in the visual representations below.

4.3.2.1 The State from t = 8:00 am to t = 8:10 am

For the purpose of testing, let's assume that the Start State of the model is at t = 8:00 am. The reporting structure described in the scenario is visually depicted in Figure 4.2a with node H, node A, and node B owning the Information pieces A_L, B_M, and H_O, respectively, in their physical Information Banks. The aggregate of the head node H logically houses all the Information of this sub-department.

The State at t = 8:00 am is represented as: $S_{8:00}$

After 10 minutes, at t = 8:10 am, node A edits the case file, which is Information piece A_L in the SIF model. We know from section 4.2.3 that editing the Information changes only the Core Information and not the Information Attributes. Hence, this Event is represented by $L_{A\lambda}$, again as explained in section 4.2.3.

Now, as the Core Information of A_L changes, it causes a State Transition from the State $S_{8:00}$ to the State $S_{8:10}$.

The complete set of Current State, Transition Event, and the next State is captured by this representation:

$$\delta\psi = (S_{8:00}, L_{A\lambda}) \rightarrow S_{8:10}$$

Below is the visual representation of the Current State and the next State. The components that changed, causing the Transition Event, are in bold letters.

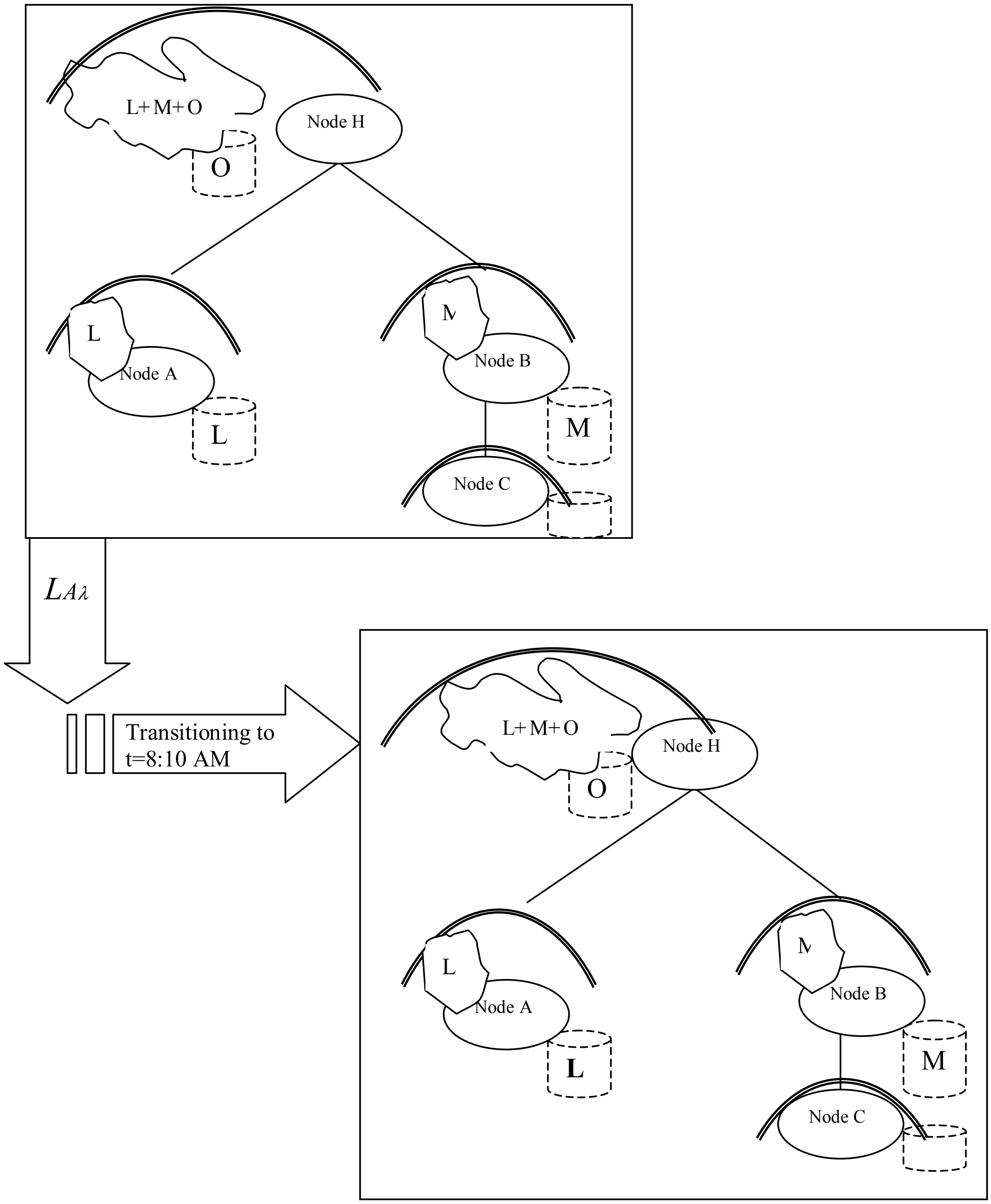


Figure 4.2a: The State at t = 8:00 am and Figure 4.2b State at t = 8:10 am

4.3.2.2 The State from t = 8:10 am to t = 8:20 am

Advancing from the previous transition, the current State is t = 8:10 am. The reporting structure, physical Information Banks, and the Information Aggregates remain the same except for the fact that they now house the changed $L_{A.}$

The State at t = 8:10 am is represented as: $S_{8:10}$.

After 10 minutes, at t = 8:20 am, node B captures new Information, which will be called $P_{B.}$ As outlined in section 4.2.3, capturing the Information created new Core Information, as well as new Information Attributes. Hence, this Event is represented by $P_{B\sigma\lambda}$.

Now, as the new Information of P_B is captured, it causes a State Transition from the State $S_{8:10}$ to the State $S_{8:20}$.

The complete set of Current State, Transition Event, and the next State is captured by this representation:

$$\delta_{\psi} = (S_{8:10}, P_{B\sigma\lambda}) \rightarrow S_{8:20}$$

Below is the visual representation of the Current State and the next State. The components that changed causing the Transition Event are in bold letters.

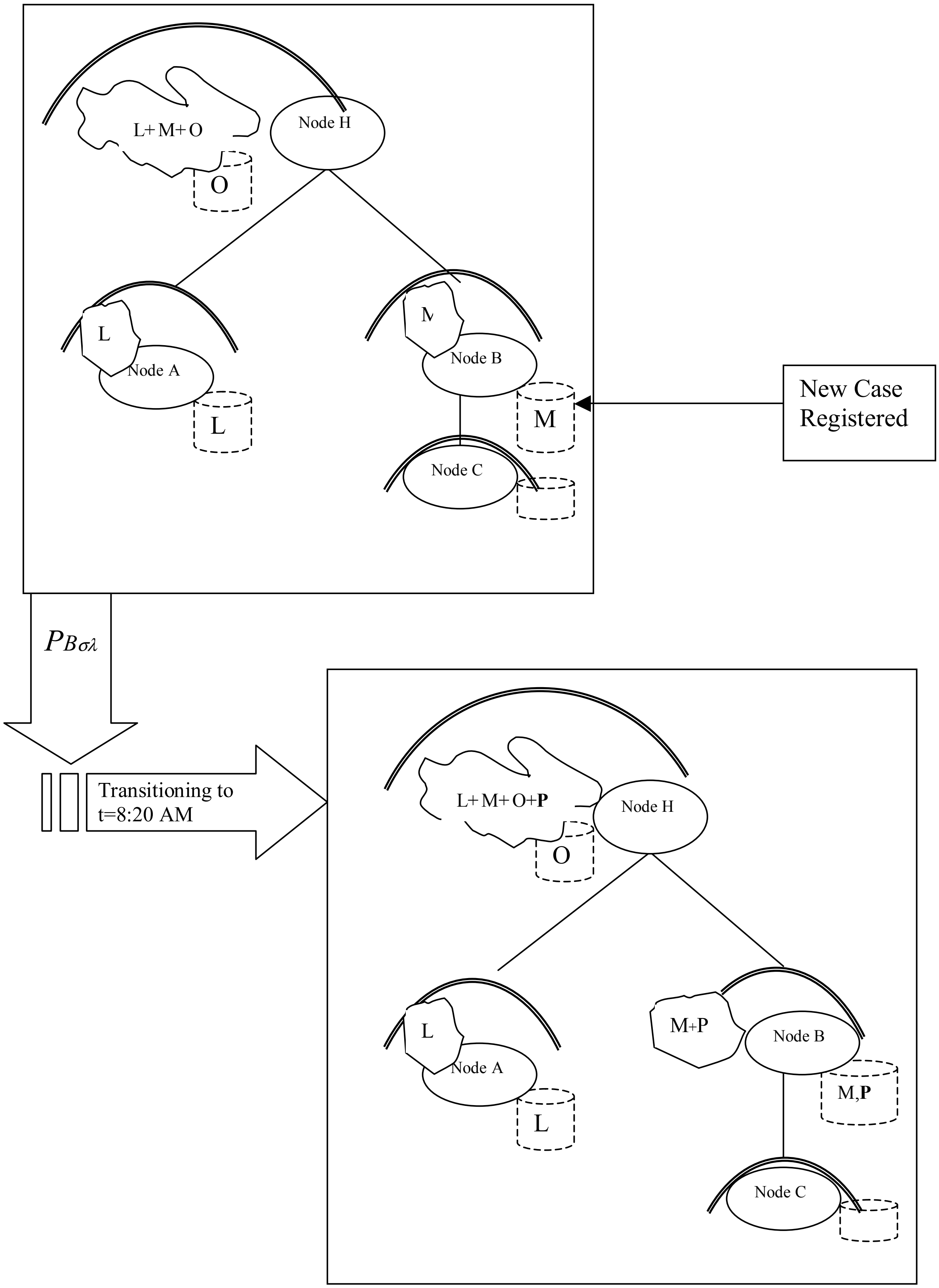


Figure 4.3a: The State at t = 8:10 am; and Figure 4.3b State at t = 8:20 am

4.3.2.3 The State from t = 8:20 am to t = 8:35 am

The current State is t = 8:20 am, as indicated by the previous transition. The reporting structure, physical Information Banks, and the Information Aggregates remain the same as the t = 8:10 State, except for the fact that new information P_B now resides in the physical Data Bank of node B, which is reflected in the senior node H's Information Aggregate.

The State at t = 8:20 am is represented as: $S_{8:20}$.

After 15 minutes, at t = 8:35 am, node B extends the Ownership right on B_P to node C. As detailed in section 4.2.3, extending the Ownership Right changes the Information Attributes. Hence, this Event is represented by $P_{B\sigma}$. However, after belonging to node C, the Information will be renamed as C_P.

Now, as the Information attribute of B_P is changed as a result of ownership right extension, it causes a State Transition from the State $S_{8:20}$ to the State $S_{8:35}$.

The complete set of Current State, Transition Event, and the next State is captured by this representation:

$$\delta_\psi = (S_{8:20}, P_{B\sigma}) \rightarrow S_{8:35}$$

Below is the visual representation of the Current State and the next State. The components that changed causing the Transition Event are in bold letters.

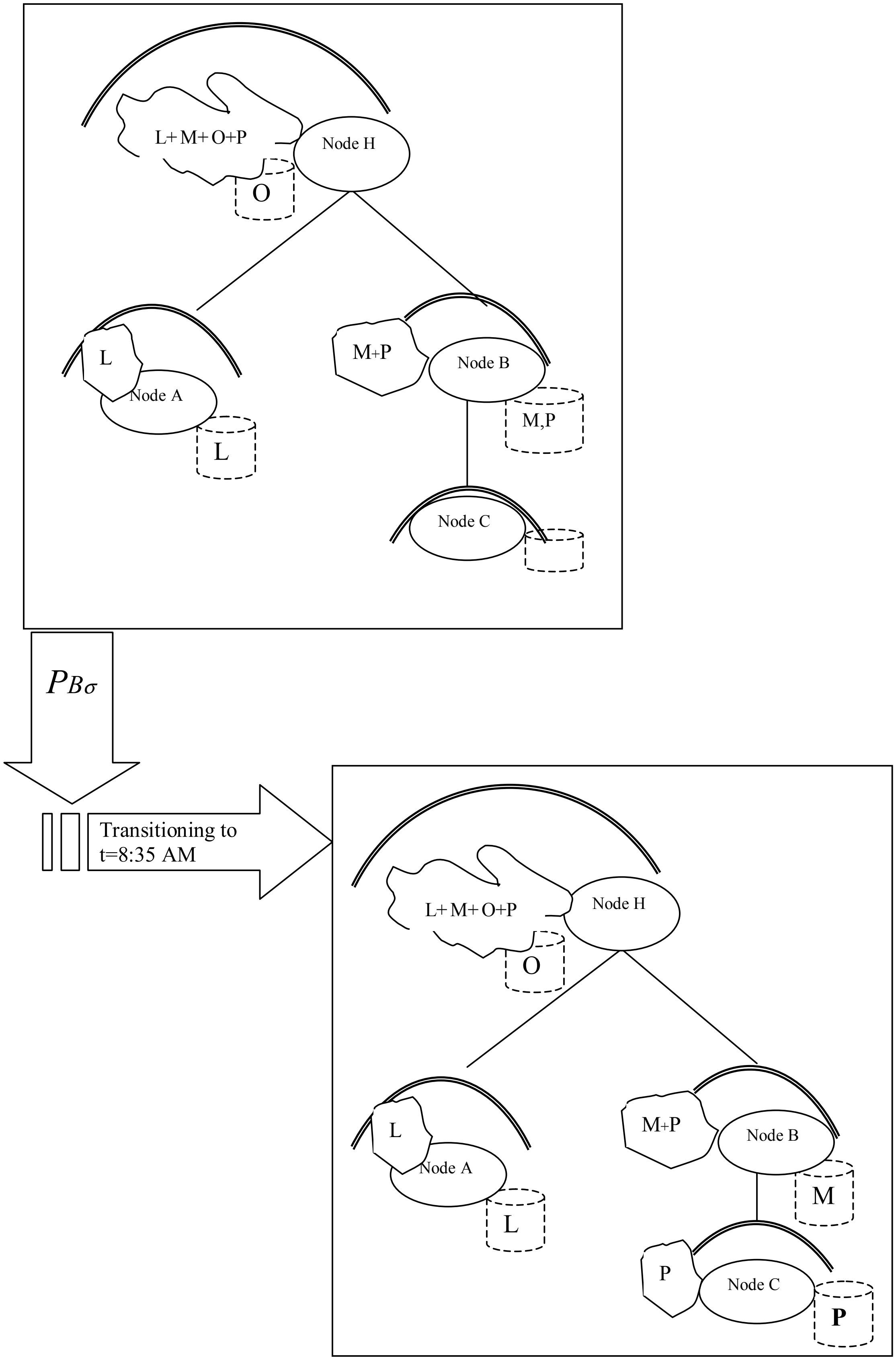


Figure 4.4a: The State at t = 8:20 am and Figure 4.4b State at t = 8:35 am

4.3.2.4 The State at t = 8:40 am

We have a description of the current State at t = 8:35 am from the previous transition. The reporting structure, physical Information Banks, and the Information Aggregates remain the same as the t = 8:20 am State, except for the fact that node C is the Owner of the Information piece P_C, which is reflected in the senior node's Information Aggregates.

The State at t = 8:35 am is represented as: $S_{8:35}$.

After 5 minutes, at t = 8:40 am, node B extends the View right on C_P, which is now in subordinate node C's physical Information Bank, to node A. As explained in section 4.2.3, extending the View Right changes the Information Attributes. Hence, this Event is represented by PC_σ.

Now, the extension of view rights causes a State Transition from the State $S_{8:35}$ to the State $S_{8:40}$.

The complete set of Current State, Transition Event, and the next State is captured by this representation:

$$\delta_\psi = (S_{8:35}, PC_\sigma) \rightarrow S_{8:40}$$

Below is the visual representation of the Current State and the next State. The components that changed causing the Transition Event are in bold letters.

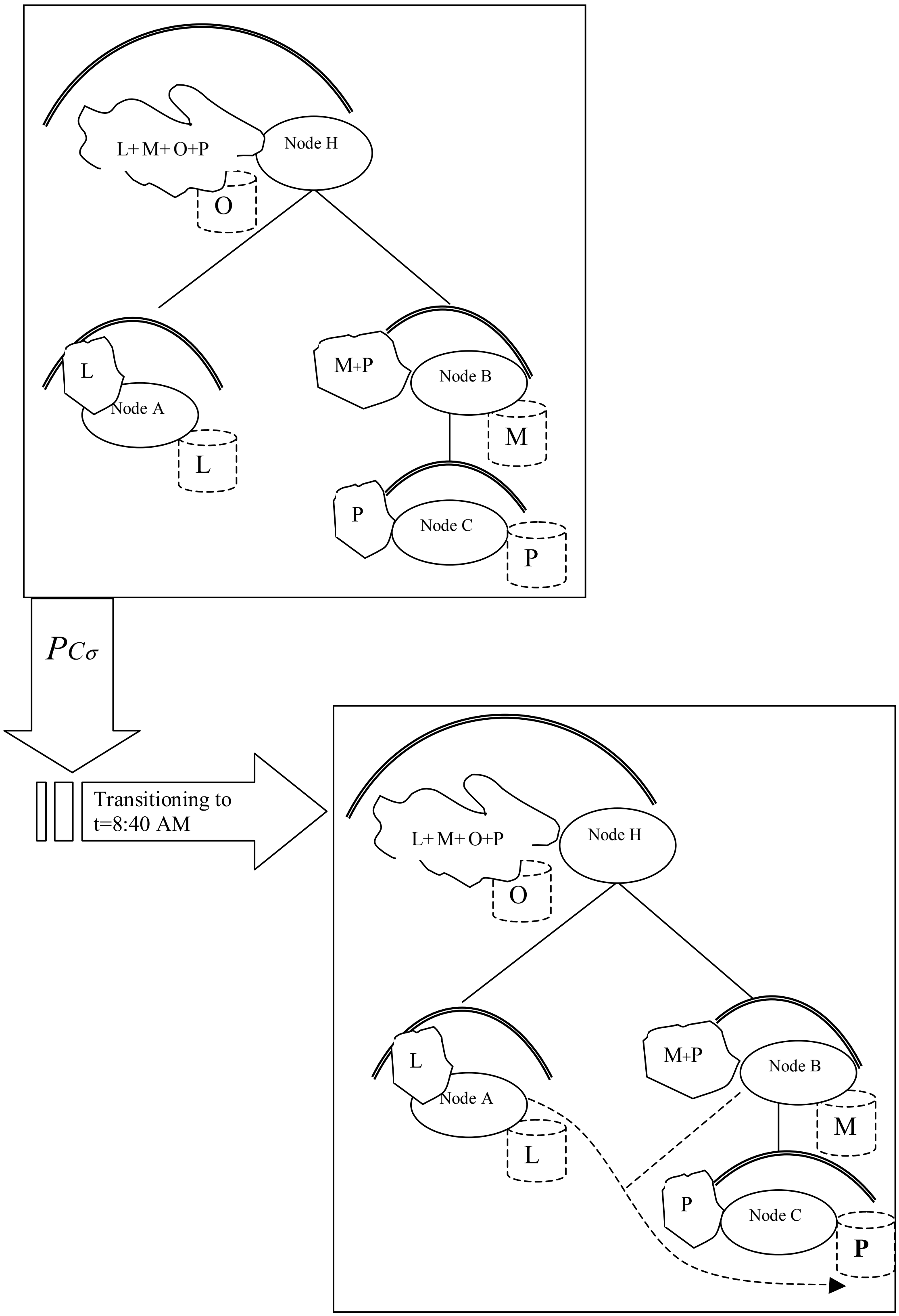


Figure 4.5a: The State at t = 8:35 am and Figure 4.5b State at t = 8:40 am

4.4 Summary

This chapter formally defines the SIF framework using a State model approach. It defines all the components of the model that constitute a State, and the events that can cause a State transition to another State. In this chapter, we discussed two candidate notational languages, then developed a hybrid notational language to efficiently capture the State and the transitions of SIF. This chapter also used the symbols developed throughout the previous chapters to visually represent the model at a more granular level.

These formal descriptions amount to a blueprint for the success of automated information flow in virtually any organization. At the end of this chapter, a real-world organization scenario was described and modeled successfully using the theory, notational language, and visual representation scheme developed earlier in the chapter. The components of SIF offer a simple, well-planned strategy for achieving the dream of a paperless office, and one needn't be a Don Quixote to complete the quest.

Chapter 5 – Information Interface Requirements

5 Information Interface Requirements

Chapter 3 developed the understanding that Interface is the logical layer between the node and the Information of interest. This layer holds the logic that implements the rules-based access in SIF. In this Chapter, a more detailed understanding of the Interfaces will be developed while formulating High Level Requirements for it.

Throughout this Chapter we will adhere to the following simple Requirements nomenclature scheme:

- HLRXX : Where HLR stands for High Level Requirements and XX stands for unique numeric digits. For example, the first High Level Requirement will be referred to as: HLR01.
- Sub-requirements will be uniquely denoted by HLRXX.XX, HLRXX.XX.XX, and so on. For example, if HLR01 has two sub-requirements, they will be referred as HLR01.01 and HLR01.02.

Certain requirements will be followed by logic captured in pseudo-language. Although requirements capture the functionality clearly, pseudo-language provides an easy way to concisely capture the conditional logic.

We have established the formula for denoting requirements and their logic. Before addressing the actual High Level Requirements, we must understand the special property of the Interface that is called 'Access Triggered Interface.'

<u>Access Triggered Interface</u>

In SIF, only the Interface of the Source node is activated in the act of Information access. Every node has an Interface of its own, but it is activated only when some other node tries to access the node's Information, or the Information in the node's Aggregate. This property of the Interface is known as Access Triggered Interface at Source, depicted in Figure 5.1.

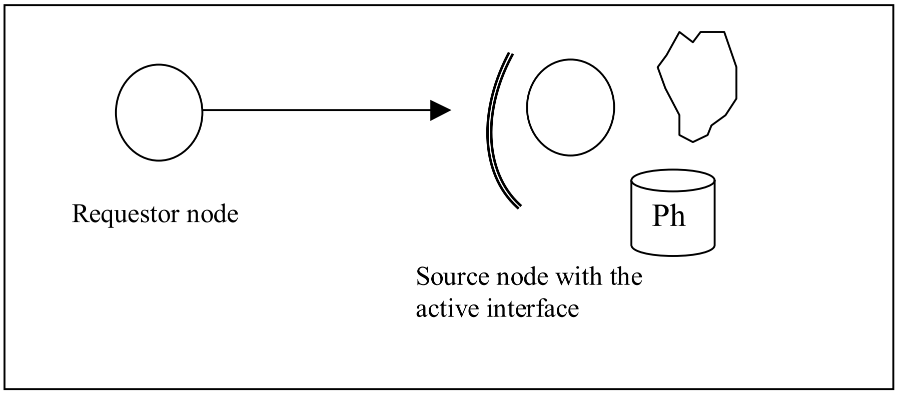


Figure 5.1: Access Triggered Interface at Source

The Information Aggregate at a node can be accessed by other nodes with appropriate access rights via the Interface. However, the node itself cannot access its own Information Aggregate. This capability would create an unnecessary redundancy in the system, since a node can always access the Aggregates of its immediate subordinates directly, rather than accessing its own Aggregate for the same information.

5.1 Information Interface Requirements

Interfaces are the mechanism to implement the rule-based Information access in SIF. It is this logic, derived from the SIF Information access rules implemented via the Interfaces, that makes the Interface intelligent. However, intelligent access is not required for all types of Information access. For some types of Information access, no rules or qualifications are required; Information can simply be released to the accessing entity. For example, any node or external entity can access the Universal Information at any time without going through any rules. Such Interfaces are considered 'dumb' Interfaces and will be referred to here as 'Non-Intelligent' Interfaces [Bayer 1996].

However, an Intelligent Interface is required for granting access on Restricted Information. Intelligent Interfaces process the rules in order to determine the access to Restricted Information.

From the above description, we can clearly identify the need for two broad types of Interface functions, which can be stated as the High-Level requirements below:

HLR01 *The Interface must provide to the requestor nodes and external entities the SIF rules-based access to the Restricted Information.*
As explained in the opening paragraph of this section, this requirement is the essence of the intelligence of the Interface. This requirement implies that the Interface must have the logic built in to process the SIF rules, and then conclude whether the requesting node can access the requested restricted Information or not. What that logic is, and what it processes, will be discussed later in more detailed requirements.

HLR02 *The Interface must provide to the nodes and external entities a direct (non-rule-based) access to the Universal Information.*
Again, as explained in the opening paragraph of this section, this requirement fulfills the Non-Intelligent functionality of the Interface. For Universal Information access, there are no rules and, therefore, no processing is required. Any node or external entity can access any piece of Universal Information any time. Interface is simply the means for the nodes and external entities to access Universal Information without introducing any logic in between.

Chapter 3, section 3.2.2.2.1, described a sub-category of Restricted Information, termed Restricted Information with External Designator. We proposed that Restricted Information with External Designator is the Information that can be accessed by deserving external entities. This qualification is satisfied by the sub-requirement below:

HLR01.01 *The Interface must allow the authenticated*[2] *external entities to access the Information designated to be accessed by the external entities.*
The High Level Requirement HLR01 provides the functionality that the Interface will give rule-based access to the external and internal entities trying to access the Restricted Information. However, unlike internal entities called nodes, there is no pre-identified set of external entities to which the Restricted Information access will be limited. There is, therefore, a need to identify the external entity that wishes to access Restricted Information, and that process is called Authentication. This requirement specifically and clearly addresses the functionality needed to provide appropriate Restricted Information access to the external entities.

As an addendum to the above requirement, the Interface will follow the logic captured in the pseudo-language below to determine the access by the external entity.

Where the Rie_Attribute represents the External Designator attribute (Rie) that contains the identity of the external node that has view right over this Information. External_id is the identity of the external node that is attempting to View the Information. Allowed and Not_Allowed indicate whether the external entity can View the Information or not. The Reverse Arrow denotes that the value on the Right Hand Side is present in the value on the Left Hand Side.

For an authenticated entity outside of the Domain:

```
If Rie_Attribute ← External_Id
View = Allowed;
        Else
        View = Not_Allowed.
```

To put it simply, the equation above means that the Information View access operation should be allowed for an external entity only if it has the View right registered in the Attribute of the Information.

---

[2] Authentication refers to any appropriate step that ensures that the external entity receiving the information is the one for whom the information is intended. For example, password access to the Information could be a means of authentication.

HLR01 and HLR02 are the two most basic requirements. In order to develop relatively detailed requirements while still remaining at a high level, it is essential to present a more granular view of Information Access operations that the SIF Interface supports.

5.2 Information Access Operations

Information Access Operations are the categories of access that a node can perform. For example, a node may edit the Information it owns or it may view the Information of another node. All the Information Access Operations by nodes take place through Interfaces. Different categories of Access Operations by nodes impose different types of requirements on the Interface. Below, Information Access Operations by a node are presented.

In total, there are three types of Information access operations in SIF that a node can perform:

1. Universal Information Access Operation – A node accessing the Universal Information
2. Self Information Access Operation – A node accessing its own physical data bank
3. Inter-node Information Access Operation – A node accessing other node's Information

Each Information Access Operation and the Interface requirement corresponding to the operation are discussed below. The operations are also depicted in Figure 5.2.

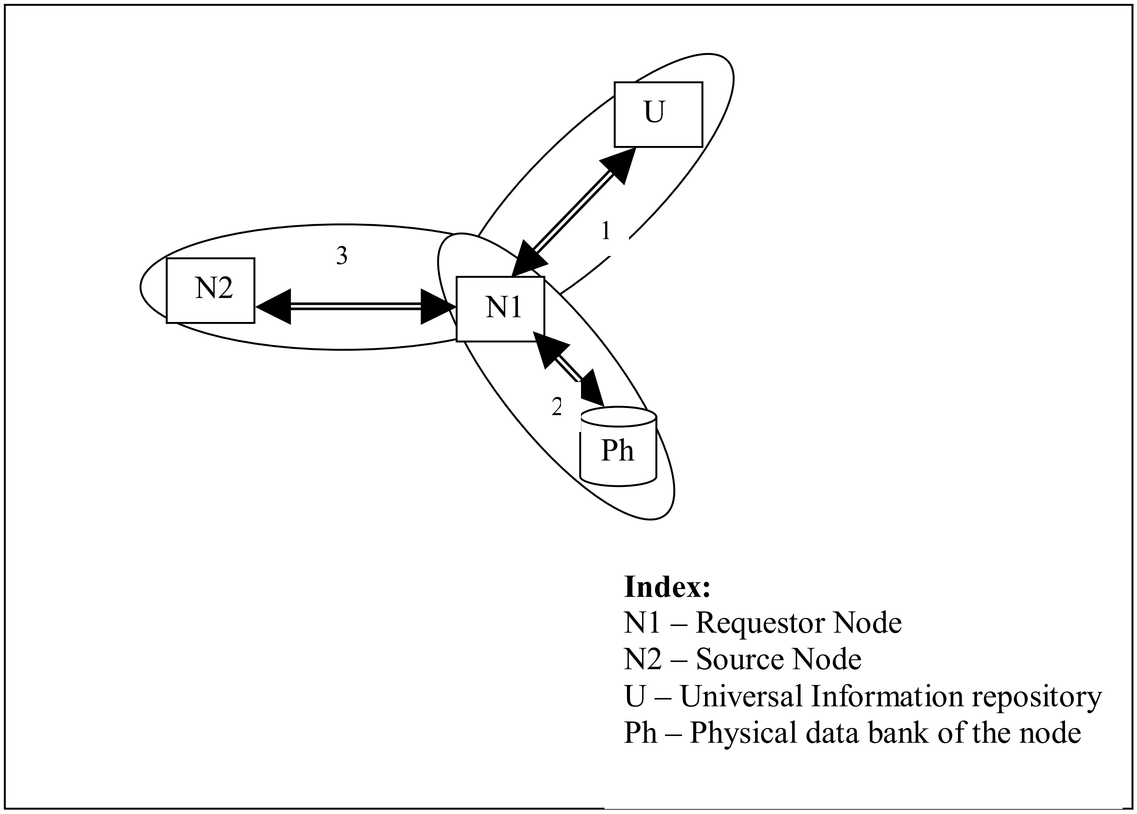


Figure 5.2: Broad Category of Information Access Operations in SIF

Universal Information Access Operation

This is the most straightforward of the Access Operations. Any node or, for that matter, any external entity can access the Universal Information without requiring any rights or privileges. This Access Operation is already covered by the requirement HLR02 for Non-Intelligent Interface, so there is no separate requirement for it.

Self Information Access Operation

This type of Information Access Operation takes place when a node needs to edit/delete its own restricted Information. Information owned by the node resides in its Physical databank. An Intelligent Interface, in order to enable the Owner to access and edit/delete its Information, must be able to perform two functions. First, the Interface must be able to determine if the requested Information piece is owned by the requestor node. Second, the Interface should display[3] the Information piece to the owner. Once the Information that is requested by the owner is identified and displayed to the owner, it is the responsibility of the owner to use appropriate software to edit/delete the requested Information piece.

Both the functions mentioned above are manifestations of SIF rules for Restricted Information access. Hence, these requirements are logically the sub-requirements of HLR01. The above two statements can be framed as the following requirements:

HLR01.02 *The Interface must be able to ascertain if the node requesting the Information from the databank with the intent of editing/deleting is the owner of the Information.* This requirement can be explained best in conjunction with the next requirement. The combined explanation of both the requirements follows the next requirement.

HLR01.03 *If the requestor node is the owner of the Information, the Interface should display the Information to the node, allowing it complete authority to edit/delete the Information.*

The above two requirements add the level of detail to the rules-based access alluded to in HLR01. Accessing the Restricted Information is a two-step process. First, a node requests access to a certain piece of Restricted Information. Second, based on whether the node is the owner or not, the node gets limited or complete access over the Restricted Information piece. HLR01.02 processes the node to determine if it is the Owner of the Information. HLR001.03 grants complete access to the Owner of the Information, which includes the right to edit and delete. If the node is not the owner of the Information, the response detailed in the requirements below ensues.

Inter-node Information Access Operation

A node vested with appropriate rights and privileges can view another node's restricted Information via Information Aggregates. Said another way, a senior node can

[3] Display implies showing the Information heading via the Interface. Node will however need to activate appropriate Software to edit or delete the Information.

view a subordinate's Information by accessing the subordinate's Aggregate. However, only an owner can edit/delete his own Information. We know that owned Information resides in the owner's physical data bank, the Interface requirements for which are covered in the section above. In this section, we develop the requirements for Intelligent Interface supporting Inter-node Information Viewing functionality.

A node can access another node's restricted Information under only two circumstances:
a. The requestor node is senior to the owner node.
b. The requestor node has View rights on the owner node's Information.

It has been established in Chapter 3 that the data regarding the hierarchical standing of the owner of the Information is stored in Information Attributes. Also, the identities of the nodes that have View rights on the Information are stored in Information Attributes, as well. At the requirements level, Interface needs to make the decision whether the requesting node can view the Information. At the detailed level, the Interface will make the decision using the data stored in the Information Attributes. This function of Interface, granting view access based on rules, is a direct derivative of the HLR01. The function can be captured as the following requirement:

HLR01.04 *The Interface must grant or deny View access to the requestor node based upon the rules.*
If the requestor node is not the Owner of the information, the maximum right the requestor node can exercise over the Information is the View right. However, whether the requestor node can even View the Information is based on Hierarchical standing and Extended rights the node enjoys, which is detailed in the requirements below.

A set of sub-requirements can define explicitly when an Interface should grant View access and when it should not:

HLR01.04.01 *The Interface shall grant the View access if the hierarchical standing of the requestor is greater than that of the source node.*
This requirement is derived from the basic rule discussed in the Chapter 3 that any superior node can View the subordinate's information. The Interface should be able to process the Hierarchical standing of the requestor node versus the Owner node where the Information resides, in order to determine if the requestor should have View access to the Information piece or not.

HLR01.04.02 *The Interface shall grant the View access if the hierarchical standing of the requestor node is NOT greater than the source node BUT the requestor node has the right to view the source node's particular Information piece.*
Another rule discussed in Chapter 3 was that a senior node can extend Viewing right to a subordinate's Information to another yet subordinate node. Similarly, an Owner node also has the power to extend the Viewing rights over its Information to any other node. This requirement states that, if a node that has an extended View right over the Information, that node should be granted the View access.

HLR01.04.03 T*he Interface must deny the requestor node the View access if the hierarchical standing of the requestor node is NOT greater than the source node AND the requestor node does NOT have the right to view source node's particular Information piece.*
This requirement resolves the situation in which the requestor node is not the Owner nor does he have any extended right over the requested Information piece. After processing the node and discovering the absence of any right over the requested Information piece, the Interface should deny the View access to the requestor node.

The logic implied in the above three sub-requirements can be captured concisely in the pseudo-language below. Some of the symbols used below are already explained at the end of the section 5.1. The two new symbols are defined as:
The Open and Close Brackets give the Hierarchical standing of the node in between the brackets.

The Greater Than symbol indicates that the node on the left is superior to the node on the right.

```
For a Node belonging to the Domain:

If [Node X] > [Node Y]
View = Allowed;
Else
        If (View_Attribute ← X )
        View = Allowed;
        Else
        View = Not_Allowed.
```

Simply put, the above means that node X should be allowed View access to Y's Information if X is superior to Y, or if X has extended rights over Y's Information.

Above are the requirements derived based on the various types of Access Operations that a node can perform. However, there are requirements that are independent of the access operations. These requirements deal with the Displaying and the Searching of the Information.

HLR01.05 *The Interface must display to the requestor node the titles of all the Information pieces to which the requestor node has access.*
Once the requestor node reaches the source node's Interface seeking the Information of interest, the Interface must display to the requestor node all the Information pieces existing at the source node to which the requestor node has access. If the Information titles displayed to the source node are limited, the node may access the Information directly or make use of the search function described next.

HLR01.06 *The Interface must allow the requestor node to search the Information of interest.*
This requirement is precursor to the function of Information access. Only if a node knows where the Information of interest exists can it request the rights or the access to that

Information. The search function mentioned in the requirement shall enable the node to search for the Information based on various search criteria like Keywords, Time Stamps, Owner, Domain, etc. The exact specifics of the search function are too detailed to be included in the High Level Requirements. A further restriction, limiting a node's search to only the Information titles to which he has access, is briefly discussed in the Future Work section in Chapter 6.

5.3 Summary

In this chapter, we discussed various categories of Information Access operations and developed an understanding of how Interface would support the Access Operations in a SIF-enhanced environment. We developed high-level requirements for Interfaces, which correspond to the functions that the Interface should support in order to implement SIF rules-based Information access. Selected requirements and sub-requirements for Interface are complimented by the logic captured in the pseudo-language defined specially for the purpose in this chapter.

Chapter 6 – Summary and Future Work

6 Summary and Future Work

In the preceding chapters, we have presented the SIF framework for Information flow automation. We began by introducing the background issues and impediments to complete Information flow automation for organizations. In Chapter 2, we gave a brief account of prior and current efforts at automation, and their possible shortcomings, over the past two to three decades. Chapter 3 introduced the conceptual components of SIF in layman terms, and the next chapter formally defined the SIF framework. The subsequent chapter detailed the concept of Interfaces in SIF and, in the process, developed Interface requirements. In this last chapter, we summarize the thesis and discuss the future work opportunities generated by this body of work.

6.1 Overall Summary

The SIF framework for Information flow seamlessly integrates and automates the Information flowing *within* the organization, *into* the organization, and *out* of the organization.

Some of the pre-Internet period approaches to Information flow automation we examined failed to include the customer as the source of incoming Information for service-providing organizations. This shortcoming was attributed to the lack of computer and network access for the public in the pre-Internet era. In the post-Internet era, it is mostly the smaller, more recent, e-commerce organizations that are utilizing the latest technology for fully automating the Information flow for the services they provide. The older and larger organizations seem to lag behind in this field for the various reasons discussed in Chapter 1. The SIF framework addresses the issues key to automation for such organizations by tackling the problem from four angles:

1. Topology
2. Information
3. Rules
4. Interfaces

Using these four angles, SIF forms an infrastructure that truly automates the Information flowing within, into, and out of the organization. The important point is that SIF creates this infrastructure utilizing the organizations' own roles and reporting structure and pre-existing conventional Information flow paths. Everything ranging from the rules and constraints of Information flow to the valid flow paths is defined using a formal language especially developed for this purpose. The Interfaces implement the rule-based Information flow that is at the heart of SIF. The requirements for such Interfaces are also discussed within this body of work.

6.2 Contributions

The main driving factor for developing the SIF framework is the absence of a standard and widely accepted approach for automating the Information flow of organizations. Some of the current automation approaches are outdated (pre-Internet period), some are too narrowly focused on a particular domain (e-commerce), and some exist on a purely technological level (standards for data exchange). Our hypothesis is that, for an automation approach to be generally applicable to a variety of organization types, it should be generic, simple, and robust. By defining the theory for SIF, we have broken the ground for the definition of such an Information flow approach. The contributions of this body of work are as follows:

- Conceptual Model for Information Flow

  In this body of work, we have conceptualized the Model that defines the Information flow infrastructure of large and complex hierarchical or almost-hierarchical[γ] organizations. This conceptual infrastructure enables the rules-based Information flow within the organization, as well as Information exchange with external entities. The detailed theory of such a conceptual model is defined in this body of work.

[γ] Almost-hierarchical organizations and how they can be made strictly hierarchical and receptive to SIF is discussed in Appendix A.

The theory of SIF offers, for the first time, an approach to automate the Information flow of an organization using the organization's own work/reporting structure, roles, and Information to define the automation model.

- Notational Language and Visual Representation

  The conceptual model we put forward in this body of work is formally defined using the notions of State, Transition, and Transition Events. A notational language specially developed as a part of this effort captures the states, transitions, and events. Also developed in this body of work is a set of symbols that visually represent the states, capture the transitions and events, and also identify the SIF component changed by the event. Both the Notational and Visual representations make SIF a very powerful medium to define and understand the system for automating the Information flow of an organization.

- Real Life Organization and Scenarios

  SIF is intended to be a pragmatic solution approach for Information flow and is designed with real world organizations in mind. Throughout this body of work, each concept put forward is applied against the simple but real world scenarios from a hypothetical police department. The concepts like Aggregation and Pooling stem from the bias towards producing a framework that can be successful in the real world, not just on paper.

- Testing the Conceptual Model

  Toward the end of this body of work, we have described a real world scenario and used the SIF framework to model the scenario. The scenario is inspired by the day-to-day workings of the police department. Modeling such a real-life scenario using SIF successfully tested the capability of the SIF framework's theory, notational language, and visual representation scheme, as put forward in this body of work.

- Pseudo Language and the High-Level Requirements for the Interfaces

Interfaces are the mechanism in SIF that implement the rules- and constraint-based Information flow. The logic that renders intelligence to the Interfaces is based on the rules, and is defined in this body of work using a Pseudo Language specially developed for the purpose.

While all the theory in SIF is geared towards the "What" (*What should the automation model be?*), Interfaces lean more towards the "How" (*How will the automation model work?*). In this body of work, we define high-level requirements for the Interfaces. Interfaces take the SIF framework theory a step forward by defining at a high-level the functionalities the Interface must support in order to manifest the desired Information flow to reality.

6.3 Future Work

The aim of the work presented in this thesis is to provide a framework for developing the infrastructure to automate the Information flow of organizations. SIF, as it is presented here, is a fully developed model, which can be applied to an organization with the intent to achieve total information flow automation. However, there is much room for future development of this concept:

- Throughout the thesis, a hypothetical police department is used as an example to model and explain SIF-based Information flow. Such a model can be prototyped using a simulation model, which can visually depict the flow of the Information in the organization. This prototype can also be used to assess change in the patterns of Information flow if one or many of the following are changed in the organization:
    - Number or nodes
    - Relationship between the nodes
    - Information security tag (Restricted or Universal), etc.

  Such a prototype, with appropriate data, will also be helpful in understanding the data traffic for the organization and, as well, determining the specification of the hardware if such a system was to be implemented.

- SIF mainly models the Information flow at a logical level. Future work can define SIF at a physical level, determining where the Aggregate Information would physically reside, or where the Universal Information would be housed. Defining SIF at a physical level is equivalent to providing the blueprint of implementation of a SIF-based Information flow system.

- SIF provides the conceptual model for the very basic Information Flow Infrastructure for a single organization. A topic for future work is how does SIF fit into the Service Oriented Architecture (SOA) for multi-organization Information Flow solutions? Or, more specifically, how and to what extent does the SIF conceptual model need to be modified to allow SOA infrastructures to take advantage of the SIF's approach to characterizing and implementing Information Flow?

- There are various types of Information flow rights and privileges that different entities enjoy in SIF. As detailed in SIF, however, these entities are primarily single nodes with total authority over items of information. Is there an advantage in providing such rights and privileges at a group level? Such an approach would help conceptualize rights and privileges from a more Informational perspective rather than an organizational one.

- In this body of work, we have defined that, for security reasons, an Interface can display to a node only the titles of the Information to which the node has access. This restriction substantially limits the capability of the node to search the Information of interest to which it does not already have access rights. The issue of displaying to the requestor node the Information titles to which the node does not have the access right, without causing an Information security concern, is an issue that can be tackled as a part of the future work.

- SIF provides a robust and intuitive mechanism of automated Information flow in the organization. Advanced Information flow tools, such as a Work Flow tool, that define the task, schedule, and sequence of Information flow between the nodes can be layered on top of the SIF framework.

- SIF is a generic Information flow automation framework. Its versatility and effectiveness can best be demonstrated by modeling organizations specializing in e-commerce, e-education, e-training, e-governance, and others.

SIF, as defined in this body of work, is designed to manage Information flow in organizations with hierarchical work/reporting structure. The SIF approach can be diversified to make it applicable to organizations with alternate types of work/reporting structures.

6.4 Summary

In this last chapter, we summarized the motivation and goals of this body of work. We also briefly reiterated the key components of SIF. The key contributions made to the area of research by this body of work are also described in this chapter. Finally, we discussed the opportunities of future research generated as a result of this theoretical research work.

References

Aversano L., Gerardo C., 2002, *Introducing eservices in business process models*, Proceedings of the 14th international conference on Software engineering and knowledge engineering, 2002, Vol. 27, P. 481 - 488

Baggio G., Wainer, J., Ellis, C., 2004**,** *Applying scheduling techniques to minimize the number of late jobs in workflow systems***,** Proceedings of the 2004 ACM symposium on Applied computing, 2004, P. 1396 - 1403

Bayer, S., 1996, *Embedding Speech In Web Interfaces*, Proc ICSLP, Philadelphia, PA, Oct 1996, P. 1684-1688.

Bensalem, S., Lakhnech, Y., Owre, S., 1998, *Computing Abstractions of Infinite State Systems Compositionally and Automatically*, CAV'98: Computer Aided Verification, Number 1427 in LNCS, Springer-Verlag 1998, P. 319-331

Boisot M., Canals, A., 2004, Data, Information and knowledge: have we got it right? Working Paper Series WP04-002, February 2004 Internet Interdisciplinary Institute (IN3): http://www.uoc.edu/in3/eng/index.htm

Bhaskaran, K., Jen-Yao Chung, Das, R., Heath, T., Kumaran, S., Nandi, P., 2001, *An e-business integration and collaboration platform for B2B e-commerce*, Advanced Issues of E-Commerce and Web-Based Information Systems, 2001, P. 120-122

Georgakopoulos, D., Hornick, M., Sheth, A., 1995, *An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure,* 3rd ed., Boston: Kluwer Academic Publishers, 1995.

Grudin J., 1990, *The computer reaches out: the historical continuity of interface design, Conference on Human Factors in Computing Systems***,** Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people**,** Seattle, Washington, United States, 1990, P. 261 – 268

Hammouda I., Kai K., 2002, *A pattern-based J2EE application development environment*, Nordic Journal of Computing, Volume 9, Issue 3, September 2002, P. 248 - 260

Hoffner, Y., Lwdwig, H., Gulcu C., Grefen, P., 2000, *An architecture for cross-organisational business processes,* Advanced Issues of E-Commerce and Web-Based Information Systems, 2000, P. 2-11

http://dictionary.reference.com, 2005

http://www.unix-manuals.com, 2005

Lau, T.T., 1990, *Building a paperless office with document image processing system*, Professional Communication Conference, 'Communication Across the Sea: North American and European Practices'., Guildford**,** September 12-14, 1990, P. 40 – 42

Lee, D., Yannakakis, M., 1996, *Principles and methods of testing finite state machines-a survey¸* Proceedings of the IEEE, Volume 84, Issue 8, August 1996, P. 1090-1123

Liu Z., Stork, D. G., 2000, *Is Paperless Really More?* Communications of the ACM Volume 43, Issue 11, November 2000, P. 94 – 97

Oiso, H., Komoda, N., 2000, *Development of inter-college e-commerce system through extending a legacy system,* Advanced Issues of E-Commerce and Web-Based Information Systems, 2000, P. 210-214

Patterson D., Olah, G., 1981, *Using new technology for systems documentation: Specifications for an on-line documentation system for the paperless office***,** Proceedings of the ACM '81 conference, 1981, Pages: 323 - 324

Petrakis, J. M., Engiles, M. J., 2000, *Creating a paperless municipal court*, Winter Simulation Conference**,** Proceedings of the 32nd conference on Winter simulation Orlando, Florida, 2000, P. 2029-2035

Smart, K.L., 1995, *The paperless office: facts and fictions,* Professional Communication Conference, 1995. IPCC '95 Proceedings. 'Smooth sailing to the Future'., IEEE International**,** P. 141, September 27-29, 1995, Savannah, GA

United Nations Economic Commission (UN/EC): Electronic Data Interchange for Administration, Commerce and Transport – Application Level Syntax Rules (ISO 9735), 1991.

Appendix A

We recognize that all the relationships in the reporting structure of an organization may not yield crystal clear hierarchical topology. There might be situations in a group of nodes in which certain node(s) act as senior(s) in a set of circumstances, while in another set of circumstances the same node(s) act as subordinate(s).

The simple strategy proposed in the Information Based Precedence Table, demonstrated below, establishes clear hierarchies between the nodes based on which Information is being processed in which scenario. The hypothesis is that overall hierarchy may not be established between the nodes, but there exists a clear cut hierarchy between the nodes for a given piece of Information. In other words, there is a clear precedence of nodes established over the use of the Information.

Figure A.1 shows a conflicting hierarchical structure, which can be mapping of a portion of the reporting structure of an organization. It consists of four nodes: D1, D2, D3, and D4.

In the Figure A.1 below, an outgoing arrow goes from 'reportee' to reporter. The figure interprets to following reporting structure, which is clearly self-contradictory:

D3 and D2 report to D1.

D1 reports to D4.

D4 reports to D2.

D2 reports to D3.

Assume that Information 'I' flows into and out of the given collection of nodes. Based on processing requirements, for example: Information 'I' can be broken into 4 pieces: I1, I2, I3, and I4. This scenario can easily be related to a real situation in a police department. An example might be the report of an abandoned and injured pet. That Information will be broken down into smaller pieces, which will drive the work of three sub-units of the police department, namely:

- Emergency Services – for medical help
- Veterinary Patrol – For animal abuse investigation
- Lost & Found – To check for any missing animal report

So there is an Information piece, perhaps about the location of the injured animal, that is owned by Emergency Service, but it is of use to Lost & Found and Veterinary Patrol, as well. The ensuing process will establish primary and secondary ownership on the Information piece. This is the Information-based Precedence.

After some analysis, here are the results of precedence over Information piece I1 through I4 as presented in Table A.1. Below is the simple representation of Table A.1:

D1 holds primary precedence over Information piece I1 and secondary precedence over Information pieces I2 and I4.

D2 holds primary precedence over Information piece I2 and secondary precedence over Information piece I1.

D3 holds primary precedence over Information piece I3 and ternary precedence over Information piece I2 and quandary precedence over Information piece I4.

D4 holds primary precedence over Information piece I4 and ternary precedence over Information piece I1 and quandary precedence over Information piece I2.

A dash (-) implies that that particular node has no precedence over particular Information.

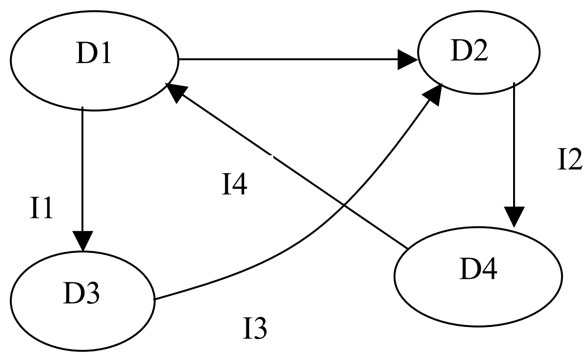


Figure A.1 : A Conflicting Hierarchy

nodes	Information Pieces			
	I1	I2	I3	I4
D1	1	2	-	2
D2	2	1	-	-
D3	4	3	1	-
D4	3	4	-	1

Table A.1 : Information Based Precedence

Once there is a clear precedence of nodes on different Information pieces, a different hierarchical topology view from each Information piece can be formed. For Figure A.1 and Table A.1, there will be four views, one for each Information piece I1 through I4, as shown in Figure A.2.

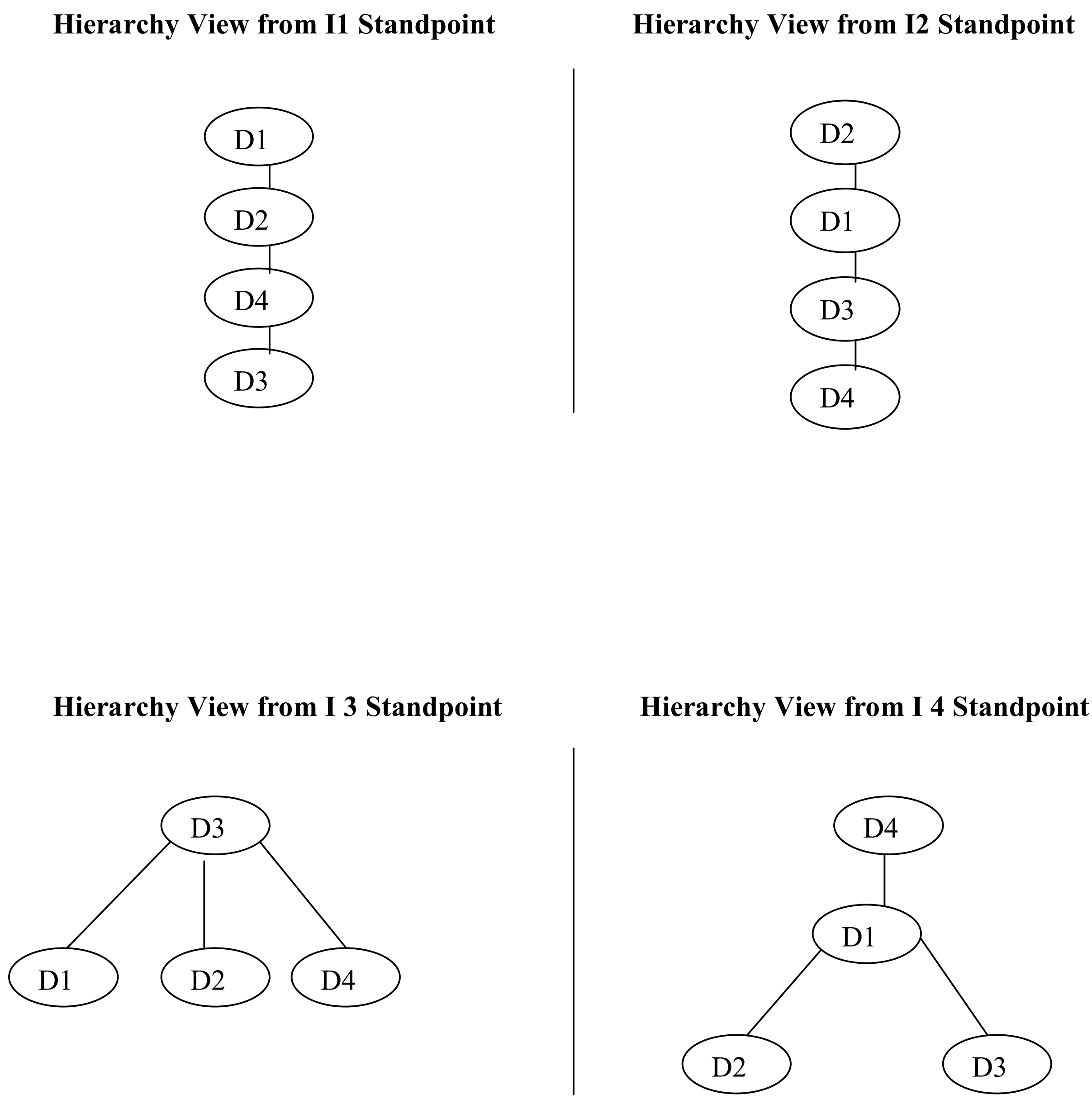


Figure A.2 : Resolved Hierarchy Views

If a SIF model were to be built using the hierarchically conflicting portion of the organization represented in Figure A.1, one of the hierarchical structures shown in Figure A.2 could be used, depending upon the Information piece under consideration. However, in order to have a complete view of the system, four views - each representing a perspective based on the standpoint of one of the Information pieces – would be required.

Appendix B

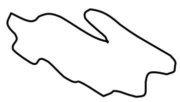
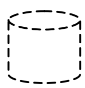
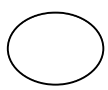
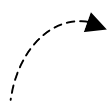
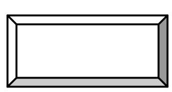
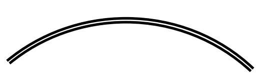
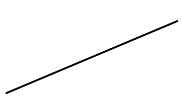
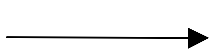
Symbol	Meaning
	Aggregate - containing the cumulative Information of the node and its immediate subordinates
	Physical Information Bank - containing Information pieces owned by the node
	node – the conceptual work processing entity
	Logical Flow of Information to update the Superior node's Aggregate
I_A+I_{C+} I_{E+} I_{F+} ...	List of Information Pieces owned by a node or listed in an Aggregate
	User Screen at each node
	The Intelligent Interface
	The Non-Intelligent Interface
	Symbolizing the work structure by connecting superior and subordinate nodes
	Requesting or Granting Information

Table B.1 : Frequently Used Symbols and Their Meanings

Vita

Shishir Bhatia was born on December 25th, 1978 in Roorkee, India. Owing to his father's transferable job as a high-profile technical educator, he lived in various places across central India.

He graduated from the Government Engineering College Bhopal (now known as Rajiv Gandhi Technical University, Bhopal) in June 2001, with a Bachelor's degree in Computer Science and Engineering. He stood as the University Topper in the final undergraduate semester.

Shishir worked as a Systems Analyst for a multi-national software consulting agency, Suvi Information Systems Pvt Ltd, for a year after finishing his undergraduate degree. He came to pursue a Master's degree in Computer Science at Virginia Tech, Blacksburg, Virginia, in 2002. After finishing his coursework at Virginia Tech and an Internship with CapitalOne, Shishir subsequently took a full time job opportunity with a premier consulting agency, Analysts International, in early 2004. He spent the next two years consulting in the IT departments of Fortune 500 companies.

While working toward his Master's degree, he has served as a Graduate Teaching Assistant, Graduate Research Assistant, Lab Instructor, and researched in the fields of Electronic Systems (eSystems) and Requirements Engineering.

On completion of his Master's degree, Bhatia will join CapitalOne IT in the capacity of a Senior Project Manager, based at Corporate Headquarters in Washington, DC. He can be reached at shbhatia@vt.edu.

Shishir Bhatia